



Sistemas Informáticos

Curso 2002-03

Implementación de un sistema para intercambio electrónico (e-Barter).

Rubén Torresano Rubio
Francisco Javier Niño Fernández
René Alonso Peña Sánchez

Dirigido por:
Prof. Manuel Nuñez
Departamento de sistemas informáticos y programación

Facultad de Informática
Universidad Complutense de Madrid



Resumen del proyecto e-Barter

El gran desarrollo que ha experimentado Internet en los últimos 20 años, y la popularidad e instauración en los hogares de todo el mundo, ha llevado cada vez más, al desarrollo de aplicaciones y soluciones para la red de redes, con el fin de hacer al usuario más fácil sus tareas cotidianas.

En los últimos años ha habido un gran auge de las conocidas como aplicaciones e-commerce, en las cuales muchas grandes empresas dedican una gran parte de sus presupuestos en la investigación y desarrollo de soluciones en este campo.

El sistema e-Barter es una aplicación de intercambio electrónico, donde los usuarios tratan de intercambiar recursos entre sí.

En la red existen muchos portales (EBAY, EBAZAR...) dedicados al poner en contacto a personas que quiere comprar un determinado producto y otras que desean venderlo.

En todo este tipo de aplicativos el usuario no se ve liberado de ninguna carga, es decir, el propio usuario debe encargarse de buscar, comparar y pujar por los productos de las distintas ofertas.

La novedad en e-Barter es precisamente la liberación de este tipo de carga al usuario, delegándola en el propio sistema, de modo que una vez introducida las preferencias del usuario, éste sólo tendrá que esperar a que el sistema le envíe la respuesta con los resultados obtenidos.

Cada usuario debe de especificar qué quiere conseguir y qué está dispuesto a dar a cambio. El sistema recoge todas las peticiones y las confronta entre sí a fin de obtener el mayor número de éxitos posible. Una vez finalizado el proceso se enviarán notificaciones a los usuarios implicados en posibles transacciones. Cada uno de ellos deberá de dar el visto bueno a la transacción en la que participan, de forma que sólo una vez que todos la hayan confirmado podrá ser realizada.

El sistema e-Barter está pensado para ser escalonado, de modo que se hacen las búsquedas de forma incremental, buscando dentro de una misma comunidad y en caso de no encontrar solución, se establece una colaboración entre las distintas comunidades para resolver las peticiones que queden pendientes¹.

El aplicativo e-Barter tiene carácter lucrativo y permite asignar un precio a cada recurso implicado en una transacción y calcular un precio final para cada intercambio propuesto por un usuario.

¹ Nota: En este proyecto sólo se han tratado todos los usuarios como pertenecientes a la misma comunidad. Sin embargo la implantación de las distintas comunidades es muy sencilla a partir del desarrollo actual.



Resume to e-Barter project

The great development that Internet has experienced in the last 20 years, and the popularity and reception in the homes from all over the world, it has taken to the development of applications and solutions for Internet, with the purpose of making easier the user daily tasks.

In the last years there has been a great peak of the acquaintances as applications e-commerce, where many big companies dedicate a great part of their budgets to the investigation and development of solutions in this field.

The system e-Barter it is an application of electronic exchange, where the users try to exchange resources to each other.

In the net there are many portals (EBAY, EBAZAR...) dedicated to getting hold of people that want to buy a certain product and other people that want to sell it.

In this whole application the user is not liberated of any load, that is, the own user should take charge of looking for, to compare and to bid for the products of the different offers.

The novelty in e-Barter it is in fact the liberation from this load to the user, delegating it in the e-Barter system, so once introduced the user's preferences, he will only have to wait to the system sends him the answer with the obtained results.

Each user should specify what he wants to get and what it is willing to give to change. The system picks up all the petitions and it confronts them to each other in order to obtain the biggest possible number of successes. Once concluded the process notifications will be sent the users implied in possible transactions. Each one of them will give the seen good to the transaction, so that only once all they have confirmed it will be able to be complete.

The system e-Barter was thought to be incremental, looking for inside oneself community and in the event of not finding solution, a collaboration settles down among the different communities to solve the petitions that are pending¹.

The application e-Barter has lucrative character and it allows to assign a price to each resource implied in a transaction and calculate a final price for each exchange proposed by an user.

¹ In this project all the users belong to the same community. However the implantation of the different communities is very simple starting from the current development.



Autorización de uso.

Rubén Torresano Rubio, Francisco Javier Niño Fernández y René Alonso Peña Sánchez desarrolladores del proyecto "Implementación de un sistema para intercambio electrónico (e-barter)", autorizamos a la Universidad Complutense de Madrid a la difusión y utilización con fines académicos, no comerciales, de todo el material desarrollado durante el proyecto incluyendo la memoria, la documentación, el código fuente y el prototipo desarrollado.

Madrid 7 Julio 2003:

Rubén Torresano Rubio

Fco. Javier Niño Fernández

René Alonso Peña Sanchez

Firma

Firma

Firma

Palabras Clave.

A continuación se detallan un conjunto de palabras clave para la calificación y posterior localización del proyecto dentro de los sistemas de información de la facultad de informática (UCM).

Español	Ingles	Comentarios
Ebarter	Ebarter	Nombre del proyecto
Intercambio	Exchange	Acción sobre la que esta basada la finalidad del proyecto (dado unos intercambios proponer unas posibles transacciones)
Intercambio electrónico	e-exchange	Intercambio electrónico. Tipo de intercambios que gestiona el proyecto
Transacción	Transaction	Resultados propuestos por el sistema (viendo una transacción como el intercambio final de unas propuestas iniciales)
Recursos	Resources	Objetos que entran en juego dentro de cada intercambio
Web	Web	Soporte por el que se pone a disposición del usuario final el sistema eBarter
Proyecto	Proyecto	Sistema realizado como proyecto de la asignatura de Sistemas Informáticos para el curso académico 2002/03



INDICE.

SECCIÓN 1 : INTRODUCCIÓN GENERAL AL PROYECTO. 9

1	DESCRIPCIÓN GENERAL DEL PROYECTO.....	10
2	ESTRUCTURA DEL SISTEMA E-BARTER	14

SECCION 2 : MOTOR DE TRANSACCIONES 16

1	DESCRIPCIÓN DEL MOTOR DE TRANSACCIONES.....	17
2	ESTRUCTURA E IMPLEMENTACIÓN DEL ALGORITMO	19
2.1	DESCRIPCIÓN DEL ALGORITMO	19
2.2	IMPLEMENTACIÓN DEL ALGORITMO.....	20
3	DIAGRAMAS DE CLASES	23
4	SALIDA DEL MOTOR DE TRANSACCIONES.....	31
5	ESTADÍSTICAS	34

SECCION 3. : BASE DE DATOS 36

1	INTRODUCCIÓN. OBJETIVOS Y ALCANCE DEL DOCUMENTO	37
2	DIAGRAMA BASE DE DATOS.....	38
3	CATALOGO-REFERENCIA DE TABLAS.....	39
4	DESCRIPCIÓN FÍSICA DE TABLAS	40
5	DESENCADENADORES Y PROCEDIMIENTOS DE ALMACENADO.	44
5.1	DESCRIPCIÓN GENERAL Y PRINCIPALES FUNCIONALIDADES.	44
5.2	CATALOGO PROCEDIMIENTOS DE ALMACENADO.	46
5.3	DESENCADENADORES.....	51
6	AMPLIACIÓN SOPORTE MULTI-IDIOMA (APLICACIÓN ADMINISTRADOR).....	52
6.1	DESCRIPCIÓN DE LA AMPLIACIÓN.....	52
6.2	DIAGRAMA DEL MODELO	53
6.3	CATALOGO-REFERENCIA DE TABLAS.....	54
6.4	DESCRIPCIÓN FÍSICA DE TABLAS.....	54
7	INSTALACIÓN DE LA BASE DE DATOS.....	56
8	APENDICES. INSTALACIÓN Y NOTAS ADICIONALES.....	57
8.1	DATABASE SCRIPT. (SQL).....	57
8.2	HERRAMIENTA DE DESARROLLO.....	57

SECCION 4 - PROGRAMA DE ADMINISTRACIÓN. (Manual de Usuario)..... 58

1	INTRODUCCIÓN.....	59
2	ACCESO Y CONFIGURACIÓN DE LA APLICACIÓN.	60



2.1	ACCESO A LA APLICACIÓN	60
2.2	DESCRIPCIÓN DE ZONAS Y CONFIGURACIÓN DE LA APLICACIÓN.....	61
3	GESTIÓN DE RECURSOS.....	63
4	GESTIÓN DE USUARIOS.....	64
4.1	BÚSQUEDA	64
4.2	MANTENIMIENTO	66
5	GESTIÓN DE INTERCAMBIOS.....	69
5.1	BÚSQUEDA	69
5.2	MANTENIMIENTO	70
5.2.1	Edición de un nuevo intercambio.....	72
5.3	PANTALLA DE DETALLES DE TRANSACCIÓN.....	74
6	MANIPULACIÓN DE LA BASE DE DATOS.	75
6.1	EDICIÓN DE TABLAS.	75
6.2	EDITOR DE SENTENCIAS SQL.....	75
6.3	GENERADOR DE DATOS ALEATORIOS.	76
7	ENLACE CON EL CLIENTE WEB	77
8	EJECUCIÓN DEL MOTOR DE TRANSACCIONES.	78
9	FICHERO DE LOG.	80
10	REQUERIMIENTOS E INSTALACIÓN DE LA APLICACIÓN.....	81
11	INFORMACIÓN ADICIONAL.	82

SECCION 5 - CLIENTE WEB..... 83

1	WEB E-BARTER	84
1.1	INTERFAZ:	84
1.2	FUENTE DE INFORMACIÓN:	84
2	ESPECIFICACIÓN DE REQUISITOS	84
2.1	SISTEMA OPERATIVO:.....	85
2.1.1	Servidor:	85
2.1.2	Cliente:	85
3	ESTRUCTURA WEB E-BARTER.....	86
4	DOCUMENTACIÓN PÁGINAS WEB	87
4.1	SERVICIO DE REGISTRO.....	88
4.1.1	Menú principal.....	88
4.1.2	Recordar password.....	89
4.1.3	Registrarse.....	90
4.2	SERVICIO DE GESTIÓN DE INTERCAMBIOS	90
4.2.1	Menú principal.....	92
4.2.2	Modificación de datos.....	92
4.2.3	Alta de Recursos	93
4.2.4	Generar intercambios.....	94
4.2.5	Listado de recursos.....	95
4.2.6	Ver los intercambios	95
4.3	PÁGINAS COMUNES.....	96
4.3.1	Global.asa.....	96
4.3.2	Error.asp.....	96



4.3.3	Talker.asp	96
4.3.4	Stylebasic.css	97
4.3.5	Adovbs.inc	97
4.3.6	Constantes_ebarter.inc	97
4.4	SERVICIO DE AYUDA	98
5	PLANTILLAS ASP	99
5.1	AMOUNTEXCHANGES.ASP	99
5.2	AMOUNTRESOURCE.ASP	100
5.3	DATALOGIN.ASP	101
5.4	DATANEW.ASP	102
5.5	DATARESOURCE.ASP	103
5.6	DATAUSER.ASP	104
5.7	ERROR.ASP	105
5.8	EXCHANGENEW.ASP	106
5.9	LISTRESOURCE.ASP	107
5.10	LOGOUT.ASP	108
5.11	LOSTPASWW.ASP	109
5.12	LOSTPASWW2.ASP	110
5.13	PASSW.ASP	111
5.14	REQUEST.ASP	112
5.15	SHOWEXCHANGE.ASP	113
5.16	TALKER.ASP	114
5.17	VIEWEXCHANGE.ASP	115
5.18	VIEWUSER.ASP	116

SECCION 6. : MANUAL DE USUARIO WEB117

1	INTRODUCCIÓN	118
2	REGISTRO:.....	118
3	LOGIN.....	119
4	MENÚ PRINCIPAL	121
4.1	MENÚ SETTINGS	121
4.2	MENÚ EXCHANGES	121
5	AGENTE DEL SISTEMA	122
6	PANTALLA DE DATOS	123
6.1	DATOS PERSONALES	123
6.2	DATOS DE ACCESO	123
6.3	RECURSOS	124
7	INTERCAMBIOS (EXCHANGES).....	125
7.1	NEW EXCHANGE	125
7.2	VIEW EXCHANGES	126
7.3	VIEW USER	127

SECCION 7. : ANEXOS.128

1	REQUISITOS DEL SISTEMA.	129
1.1	SERVIDOR:	129
1.2	CLIENTE:	129
2	MANUAL DE INSTALACIÓN DEL SISTEMA EBARTER.....	130
2.1	DESARROLLO DE LA INSTALACIÓN	130



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Título Documento : Memoria del proyecto e-Barter

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Proyecto S.I. / e-Barter. (Año 2002/03)

Sección :

Página : 8/133

2.1.1	Sistema Operativo:	130
2.1.2	Internet Information Server (IIS5)	130
2.1.3	ASPMail.....	131
2.1.4	Configuración de Global.asa.....	131
2.1.5	SQL Server.....	131
2.1.6	BDE Delphi.....	131
2.1.7	JDK 1.3	131
2.1.8	Setup_eBarter_2000.exe	131
3	GLOSARIO DE TÉRMINOS.....	132
4	BIBLIOGRAFÍA	133



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter
Descripción : Información general sobre el proyecto eBarter.
Estructura y conceptos generales.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 1: **INTRODUCCIÓN GENERAL** Pagina : 9/133

SECCIÓN 1 : INTRODUCCIÓN GENERAL AL PROYECTO.

1 Descripción general del proyecto.

El sistema e-Barter trata de dar un paso más en los sistemas de intercambio electrónico que hay actualmente, intentando liberar al usuario de la carga que supone buscar las ofertas que mejor se adaptan a sus deseos y el posterior seguimiento de cada una de esas ofertas, haciendo ofertas y contraofertas hasta alcanzar un acuerdo.

e-Barter basa su funcionamiento en que el usuario delegue toda esta responsabilidad en un agente que le representará frente a otros agentes en un mercado virtual, donde a cada agente se le dará la oportunidad de preguntar por los distintos recursos que intenta conseguir y entablar una negociación con los agentes poseedores de estos recursos.

En esta primera versión del sistema, se ha optado por buscar el bien colectivo frente al bien individual, de forma que todos los agentes ayudan a otros agentes a completar su misión. En versiones posteriores se podría añadir al agente representante cierta personalidad del usuario representado, de forma que se comporte de forma similar a como lo haría el propio usuario.

La siguiente figura muestra los pasos que se dan en el proceso de búsquedas de transacciones:

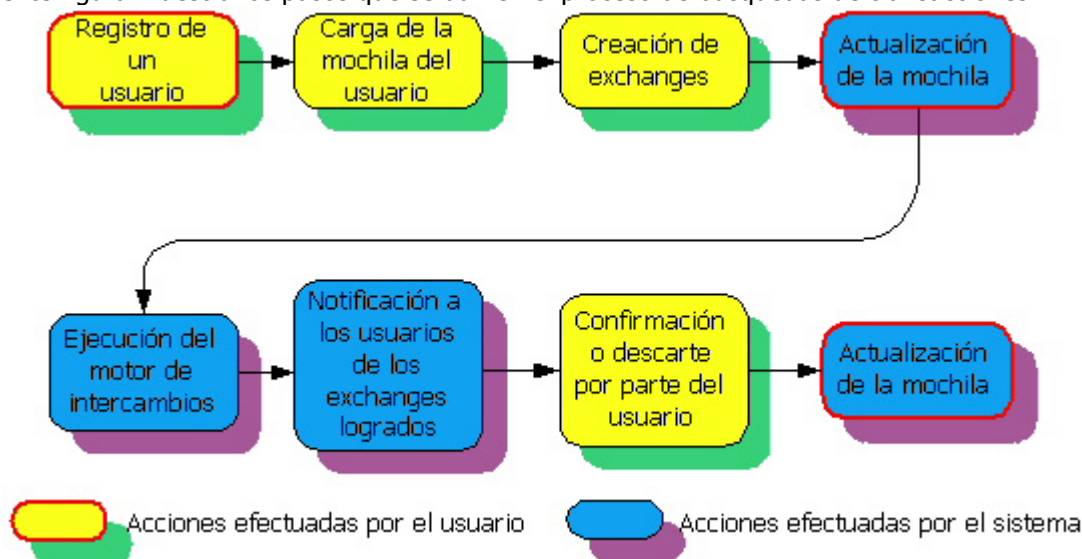


Fig. 1: Proceso por acciones

Una vez que un usuario se registra en el sistema, debe de incluir recursos en su mochila, los cuales utilizará como moneda de cambio por lo que quiera conseguir.

Posteriormente puede crear *exchanges*¹ para definir lo que desea conseguir y que estaría dispuesto a dar como máximo. Además puede crear varios relacionados de modo que en caso de lograr uno de ellos no intente conseguir los que tenía como relacionados.

Una vez creado un exchange o varios relacionados se actualiza la mochila del usuario de modo que los recursos implicados en un exchange se retienen para no poder ser utilizados en otros.

Cuando el sistema envíe al usuario la notificación de éxito para algún exchange, se le dará la posibilidad de rechazarlo y volverlo a poner en juego o eliminarlo, y la posibilidad de aceptarlo y tras la confirmación de todos los usuarios afectados se actualizaría la mochila con los recursos recién obtenidos y los que sobraron del exchange.

Los recursos representan los productos que se pueden ofrecer y conseguir en el sistema. Estos recursos se agrupan en categorías en forma de árbol sin profundidad fija. Cada recurso recibe un coste que permitirá

¹ Ver glosario.

calcular el valor final de una transacción para cada usuario, en función de los recursos obtenidos en la transacción. Actualmente, para el calculo se está usando la siguiente función:

$$\frac{\sum_{\text{recurso} \in \text{Lista de recursos conseguidos}} \text{coste_del_recurso} * \text{cantidad_de_recurso_obtenida}}{100}$$

Cada exchange está formado por una lista de recursos que se desea conseguir y otra de recursos que se está dispuesto a ofrecer a cambio. Durante el proceso, un exchange puede pasar por distintos estados, los cuales marcan las distintas líneas de acción que pueden seguir.

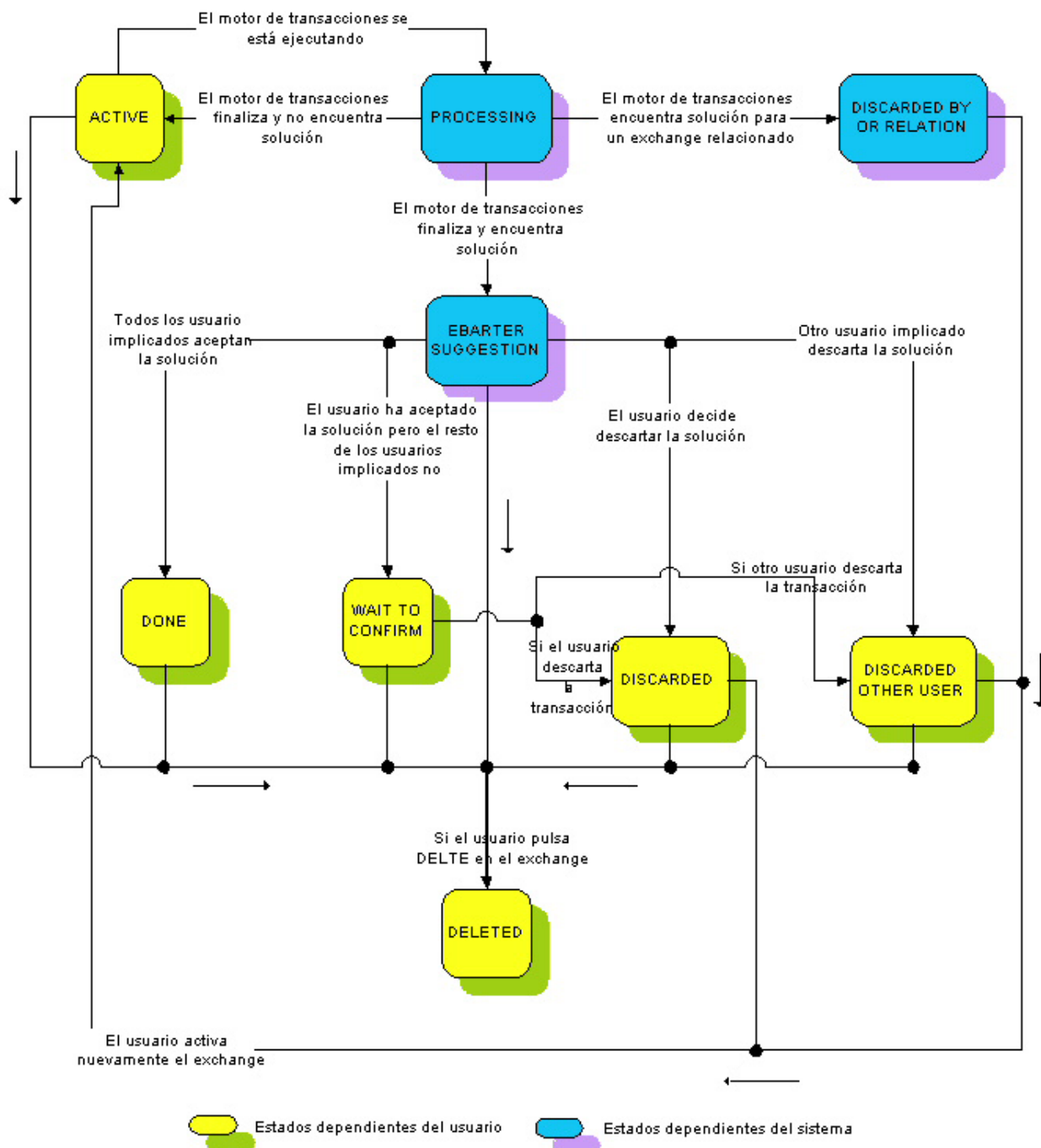


Fig. 2: Diagrama de estados de un intercambio

Ahora vamos a analizar cada una de las acciones que se realizan dentro del sistema y quien las realiza. La siguiente figura muestra esa interacción entre las distintas tareas.

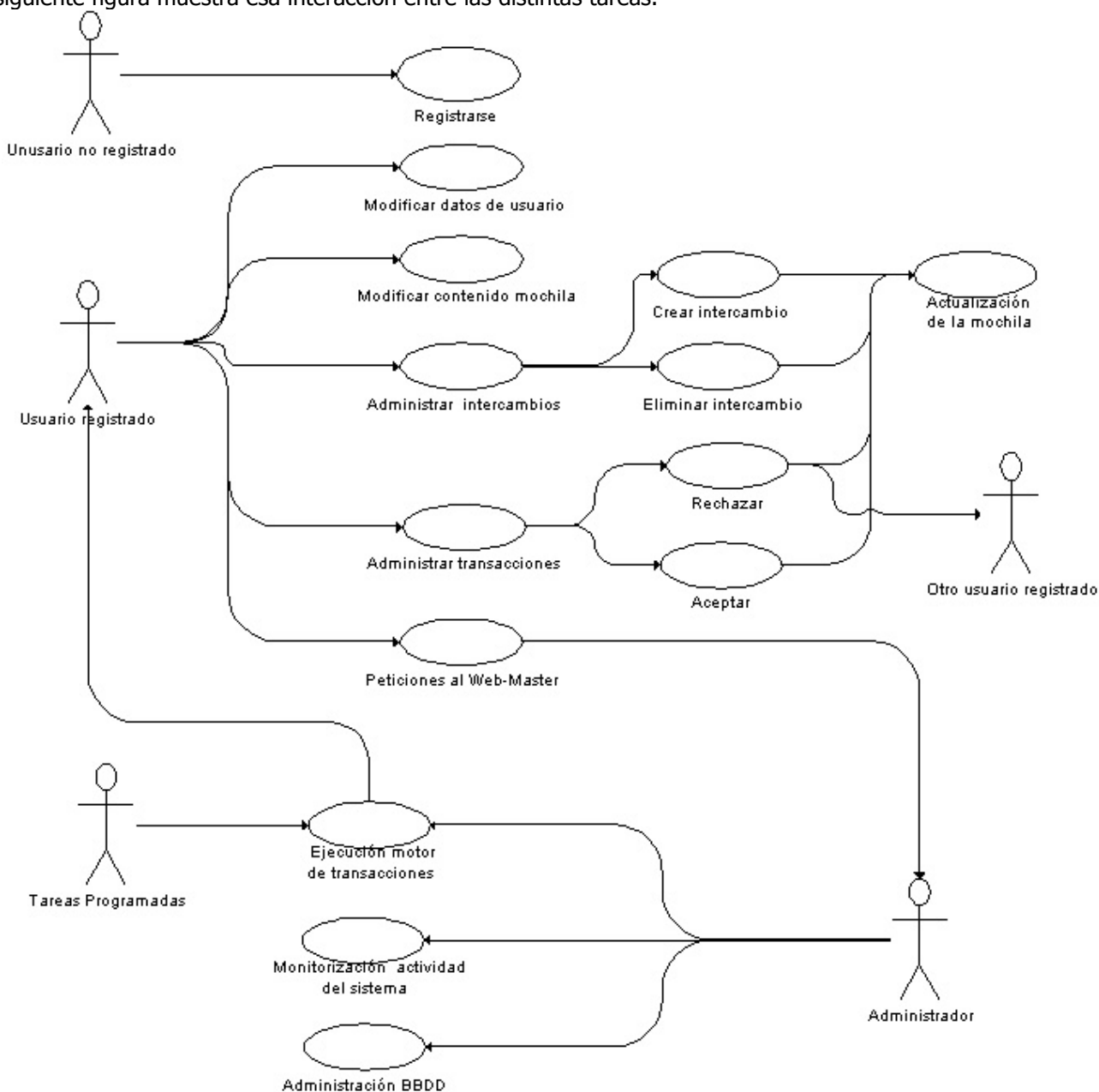


Fig. 3 Diagrama de casos de uso.

- Registro:
Para que un usuario pueda utilizar el sistema se le requerirán algunos datos para su gestión, como nombre, dirección, e-mail.....
Un usuario también tiene asociado un estado, generalmente activo. Este estado puede variar al bloqueado en caso de que el usuario no cumpla con las condiciones del portal, para el buen funcionamiento de éste, o a estado cancelado si el usuario quisiera darse de baja del servicio.
- Modificación de los datos del usuario:
El usuario podrá en cualquier momento cambiar cualquier dato, tanto personales como de acceso al portal.
- Modificar contenido de la mochila:
Para poder crear intercambios, un usuario debe de tener los recursos que está dispuesto a ofrecer en una mochila, de donde se le irán restando los recursos que ponga en juego.



- Peticiones al web master:
Un usuario puede en cualquier momento enviar un mail al web master para realizar peticiones sobre nuevos recursos a incluir, o cualquier otro tipo de asunto relacionado con el funcionamiento del portal.
- Administración de intercambios:
 - Crear intercambio:
En esta parte el usuario dice que recursos quiere conseguir y los recursos de su mochila que está dispuesto a ofrecer a cambio. Cuando se crea un intercambio, los recursos que ofreció se le restarán de la mochila, no pudiendo emplearse para otros intercambios posteriores. Además se pueden crear intercambios relacionados, de forma que en caso de lograrse uno de ellos se desestimarán los otros.
 - Eliminar intercambio:
Un intercambio o grupo de intercambios relacionados pueden ser eliminados en cualquier momento. En caso de que el intercambio aún tenga los recursos de la mochila usuario bloqueados deberá de liberarlos antes de la eliminación.
 - Activar intercambio:
Una transacción puede haber sido descartada por el propio usuario o por otro usuario implicado. En ese caso se han liberado los recursos de los intercambios afectados y se quedan en un estado de descarte en el cual no pueden volver a ser tenidos en cuenta por el motor de transacciones hasta que sean nuevamente activados.
- Administración de transacciones:
El sistema ha analizado todos los intercambios activos del sistema y ha encontrado algunas transacciones que satisfacen a algunos de estos intercambios. Entonces se procede a la notificación a los usuarios propietarios.
 - Rechazar:
Un usuario rechaza la transacción propuesta por el sistema. Todos los intercambios de la transacción se quedan en un estado de descarte liberando los recursos que tenían retenidos. Se avisa al resto de los usuarios que formaban parte en la transacción, que ha sido descartada.
 - Aceptar:
Un usuario acepta la transacción propuesta por el sistema. En este caso se pone el intercambio afectado en un estado a la espera de la confirmación del resto de los usuarios implicados en la transacción. Una vez hayan confirmado todos se procede a la liberación de los recursos sobrantes de la transacción (los que fueron ofrecidos pero no consumidos) y la actualización de la mochila con los nuevos recursos obtenidos.
- Ejecución del motor de transacciones:
Tanto el administrador como el cron¹ del sistema pueden en cualquier momento lanzar a ejecución el motor de transacciones. El motor buscará posibles transacciones entre los distintos intercambios activos en el sistema de forma que el mayor número posible de intercambios consigan los recursos que andaban buscando. Una vez finalizado el proceso se vuelven a poner como activos los intercambios que no lograron solución, los que sí lograron solución se pondrán en estado de EBARTER_SUGGESTION y los que están relacionados con intercambios logrados se pondrán en estado DISCARD_BY_OR_RELATION.
- Monitorización de la actividad del sistema:
El administrador de e-Barter puede hacer un seguimiento completo de todo lo que ha ido sucediendo en el sistema, viendo las transacciones propuestas, las contestaciones de los usuarios a dichas transacciones, análisis de los ficheros de Log generados por el motor de transacciones...
- Administración de la BBDD:
El administrador puede ver y modificar todos los parámetros y datos de la BBDD. Puede bloquear usuarios o intercambios, puede añadir recursos o categorías al sistema...

¹ Demonio de tareas programadas.

² No se recomienda modificar datos correspondientes a los intercambios de los usuarios o de las mochilas directamente en la BBDD pues pueden descompensar lo que un usuario dijo tener y lo que realmente tiene tras la modificación.

2 Estructura del sistema e-Barter

El sistema se ha dividido en tres grandes aplicativos:

- Interacción con el usuario vía Internet, mediante un portal Web. En este portal un usuario puede registrar sus preferencias, indicar las decisiones que toma frente a las sugerencias que le ofrece el sistema, etc.
- Motor de generación de transacciones. Realiza la búsqueda de las transacciones que podrían generarse atendiendo a las preferencias de los usuarios.
- Programa de auditoria y seguimiento de todo el sistema. Es un aplicativo para el administrador que permite el seguimiento de todas las acciones por parte del motor de transacciones y por parte de los usuarios.

Todo ese sistema se levanta frente a una gran base de datos donde se almacenan: usuarios, preferencias de los usuarios, recursos disponibles en el sistema, soluciones propuestas y confirmadas, registro de los archivos de Log generados por el motor de transacciones...

Además también se hace uso de un servidor SMTP para el envío de correos electrónicos a los usuarios por parte de sistema, y al administrador por parte de los usuarios.

Cada una de estas partes ha sido diseñada para no depender estructuralmente de ninguna de las otras partes, pudiendo correr todas ellas en máquinas diferentes. La siguiente figura muestra una posible estructura del sistema e-Barter:

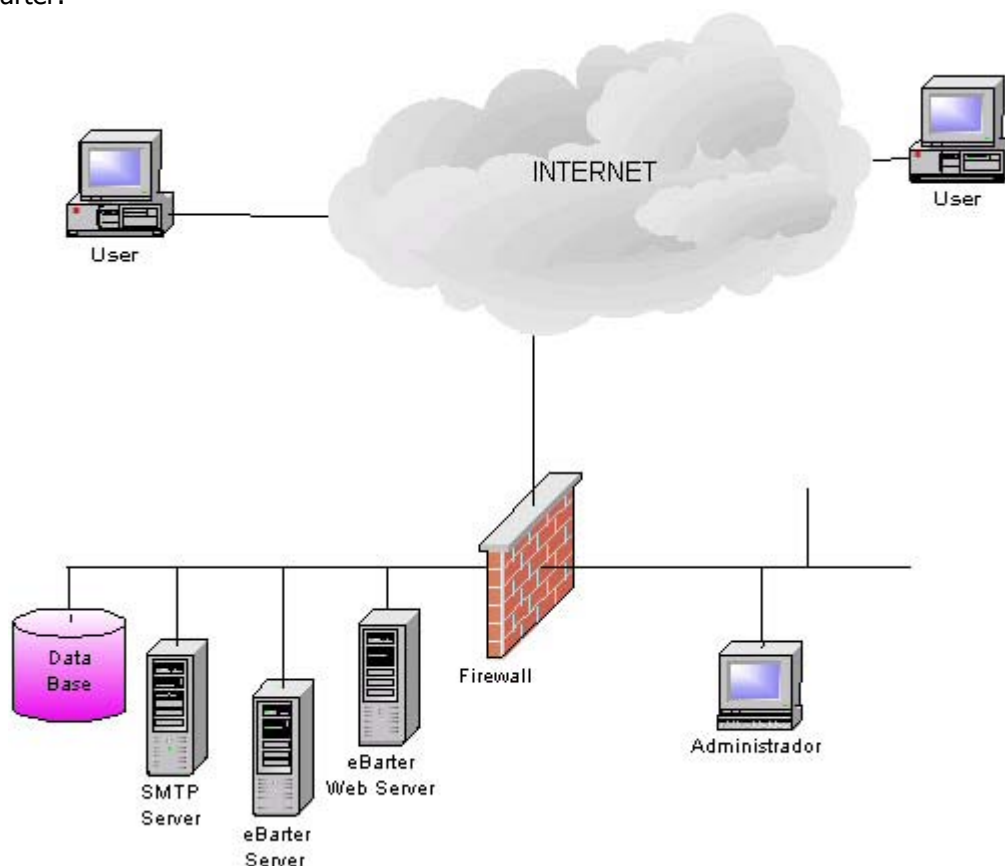


Fig. 4: Estructura de la red.



En esta estructura podemos ver los usuarios en algún punto de Internet y el sistema e-Barter en una LAN conectada a Internet por un único punto protegido por un firewall. En esta sencilla estructura tenemos al administrador y demás empleados a un lado del firewall y los servidores al otro.

La configuración del firewall debería sólo permitir tráfico entre los usuarios de Internet y el servidor Web al puerto 80 (HTTP) o 443(HTTPS)¹, además de la salida hacia Internet desde el servidor SMTP. Con respecto a la LAN debería de permitir tráfico desde la subred de empleados a la de servidores pero únicamente para la IP asignada al administrador.

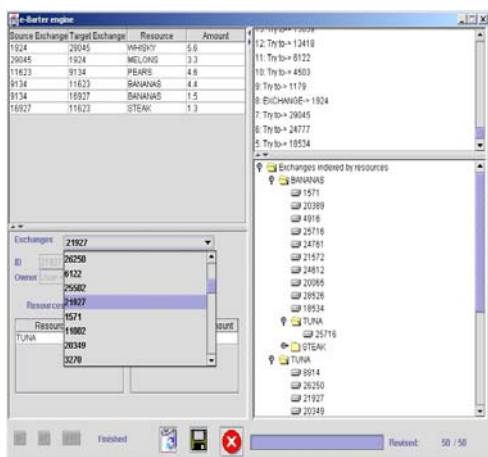
Aún con estas directrices aún se pueden realizar ataques a los servidores (principalmente a la BBDD) desde Internet utilizando los conocidos métodos de SQL Injection, y Cross side scripting (XSS). Ambos métodos aprovechan los formularios Web para incluir código malicioso que permite acceso a datos confidenciales de la y control de la máquina. Este problema afecta a todas las aplicaciones Web y no es demasiado lógico implementar una solución para corregir este problema en cada aplicación que desarrolle una empresa. Lo más lógico sería un firewall de aplicación que hiciese ese trabajo para todas las aplicaciones. Estos son sólo algunas medidas a adoptar en la implantación del sistema en un side real, sin embargo la implementación del un firewall de aplicación escapa a los objetivos iniciales del proyecto.

¹ En este caso no se ha utilizado SSL para proteger las comunicaciones y por lo tanto ese puerto también podría ser filtrado.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.
Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : **Memoria del proyecto e-Barter**
Descripción : Explicación del funcionamiento interno del proceso de generación de transacciones.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 2: **MOTOR DE TRANSACCIONES** Pagina : 16/133



SECCION 2 : MOTOR DE TRANSACCIONES



1 Descripción del motor de transacciones

El motor de transacciones es el encargado de buscar las similitudes entre los distintos intercambios de los usuarios, esta desarrollado en Java y tiene dos modos de funcionamiento:

- Modo programado sin interfaz gráfica para ser invocado desde el cron.
- Modo GUI para poder ver e ir siguiendo el proceso. Este método está más orientado a la verificación del correcto funcionamiento con ejemplos no demasiado largos, pues al utilizar Swing consume una gran cantidad de recursos para un número relativamente bajo de intercambios.

Ambos modos de funcionamiento están basados sobre el mismo motor, el cual tiene un objeto observador que va auditando todos los pasos que va dando el motor de transacciones y dependiendo del modo de funcionamiento envía los datos a un interfaz gráfico o simplemente los envía al registro de Log.

El algoritmo de esta versión del motor de transacciones se ha pensado para favorecer el mayor número de intercambios posibles de forma que los agentes colaboren entre ellos para buscar el bien colectivo por encima del bien individual.

Además se ha primado la eficiencia tanto en términos de tiempo como en resultados:

- Se ha preferido la utilización estructuras de datos complejas pero de acceso rápido (tablas hash y listas ordenadas) y utilización de varios índices de búsqueda sobre dichas estructuras a costa de un mayor consumo de memoria.
- Como se ha dicho antes se ha buscado el bien colectivo para la realización de un mayor número de transacciones.
- Se han implementado varios niveles de búsqueda de forma incremental. De esta forma se pueden producir colaboraciones entre distintos intercambios para lograr un objetivo común en el que ambos queden satisfechos. Obviamente estas colaboraciones dispara enormemente la complejidad pero ofrece un mayor índice de éxitos que las transacciones simples.
- La única poda que se ha hecho es no realizar una transacción virtual entre dos intercambios sin que al menos uno de ellos quede plenamente satisfecho. Esto es una medida para no tener que hacer backtraking tras mucho tiempo de procesamiento con el consiguiente derroche de recursos y tiempo. Además se han realizado la podas obvias como que dos intercambios relacionados no pueden tener éxito.
- Se buscado la independencia del orden de los datos de entrada en el proceso, de forma que dos ejecuciones sobre los mismos intercambios pueda producir diferentes transacciones, debido básicamente a la ordenación aleatoria de los intercambios a procesar. Con ello se pretende evitar que los intercambios más antiguos del sistema tengan mayor probabilidad de ser intercambiados, pudiendo incluso producir bloqueos. Un ejemplo se produce si a un usuario se le presenta una transacción y la descarta, si las ejecuciones siempre fueran iguales siempre se presentaría la misma solución al usuario.
- El proceso va probando distintas alternativas y cuando llega a una rama sin solución hace backtraking. A la hora de elegir una rama hay dos opciones: Hacer una copia de los objetos para en el caso de tener que hacer backtraking poder volver al estado de antes de la decisión. Otra opción es en lugar de hacer copias de todos los objetos (Exchanges, StructExchange, Barter...) llevar un array de cambios de forma que al hacer backtraking simplemente hay que deshacer los cambios mediante operaciones restore.

En este caso hemos optado por la segunda opción cuyo mayor inconveniente es los posibles efectos laterales que pueda tener un cambio del sistema. Haciendo una copia de los objetos no hay este problema pues es seguro que volverás al mismo estado que anteriormente. Sin embargo hemos considerado que el gasto de memoria sería demasiado grande.

Cuando el motor de transacciones es llevado a ejecución recoge los datos de la BBDD y los carga en memoria. Los exchanges que carga, los deja en la BBDD en estado processing para que ningún otro aplicativo pueda alterarlos.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter
Descripción : Explicación del funcionamiento interno del proceso de generación de transacciones.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 2: **MOTOR DE TRANSACCIONES** Pagina : 18/133

Finalmente, tras la finalización se ponen los intercambios implicados en alguna transacción en modo EBARTER_SUGGESTION; los intercambios relacionados con alguno implicado en una transacción se ponen en estado DISCARD_BY_OR_RELATION; los intercambios que no han conseguido su objetivo se ponen en modo ACTIVE nuevamente.

2 Estructura e implementación del algoritmo

2.1 Descripción del algoritmo

El algoritmo de búsqueda queda representado con el siguiente diagrama de flujo:

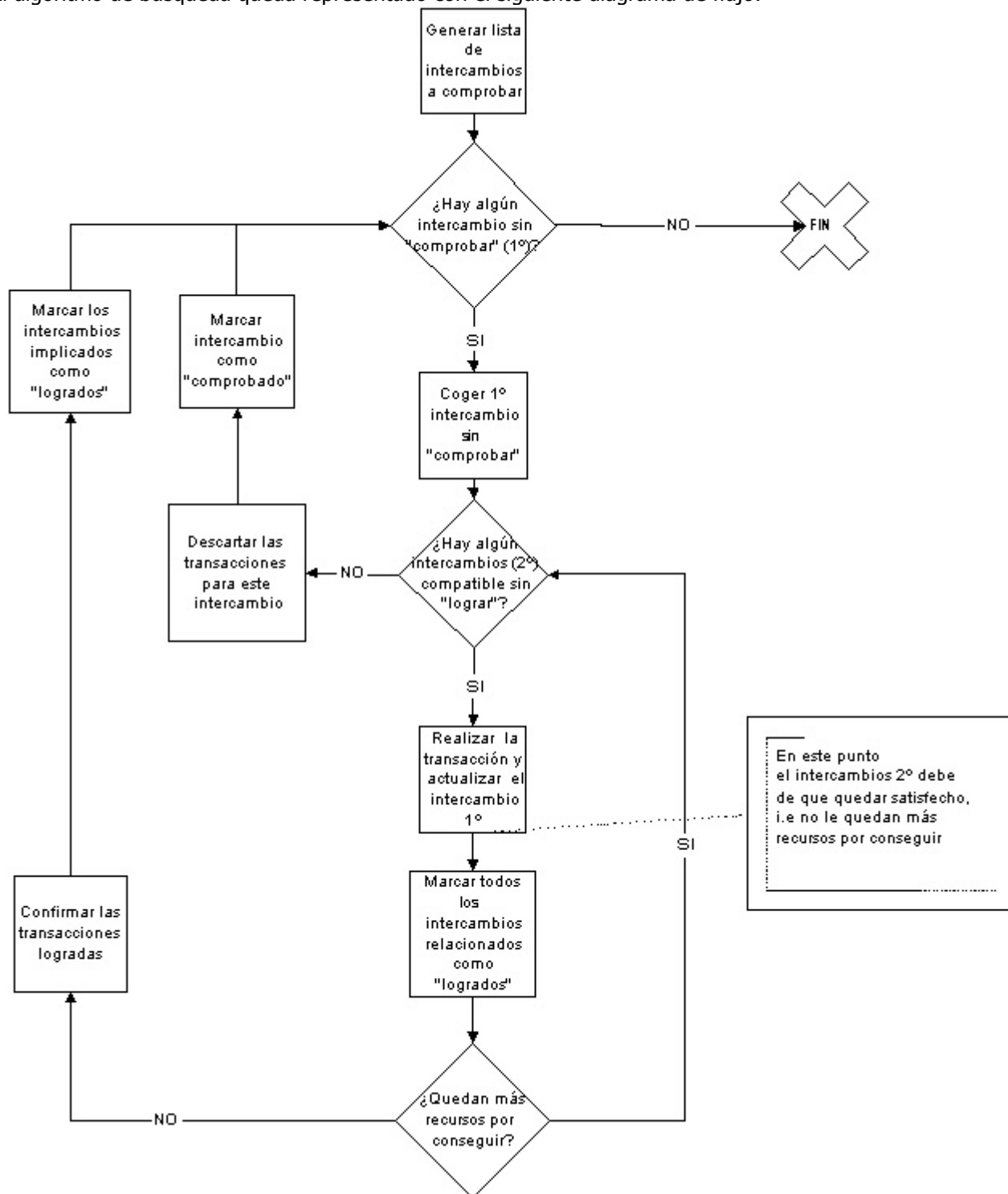


Fig. 5: Algoritmo simple de generación de transacciones.

El algoritmo marca todos los intercambios como pendientes y posteriormente los recorre todos tratando de buscar alguno que le de todo o algo de lo que está buscando. Si encuentra a alguien que le de algo de lo que está buscando se entabla la negociación. En este algoritmo es necesario que tras la negociación el segundo intercambio quede plenamente satisfecho. Si al primero aún le quedan cosas por lograr buscará a otro intercambio que le de algo de lo que le falta y a su vez éste quede plenamente satisfecho con lo que el primero le ofrezca. Si al final el primer exchange queda satisfecho se confirman todas las transacciones virtuales creándose una transacción.

Se repetirá el proceso con el siguiente intercambio por lograr.

Este algoritmo no contempla las colaboraciones entre intercambios. Para llevar a cabo estas colaboraciones se ha modificado un paso de algoritmo quedando el resto exactamente igual.

Esta modificación consiste que en lugar de coger un intercambio 1º se coge una colaboración de intercambios pendientes que consiste en unir los recursos que tienen todos y se aplica el algoritmo como si fuera un intercambio simple. Finalmente si el intercambio compuesto tiene éxito se produce el reparto de las ganancias entre los miembros cooperantes.

En esta implementación no se ha aplicado ningún tipo de poda a la hora de crear las colaboraciones, es decir cualquier intercambio puede colaborar con cualquier otro intercambio con la excepción obvia de que dos intercambios relacionados no pueden colaborar.

Hay que tener en cuenta que las colaboraciones disparan la complejidad del algoritmo. Posteriormente veremos unos datos numéricos de este echo.

2.2 Implementación del algoritmo

Para la implementación de este algoritmo hay dos claves:

- Una estructura recursiva basada en tablas hash para la mejora en los tiempos de búsqueda.
- La ordenación de los recursos en base a un peso. En este caso nos servimos en el identificador de cada recurso en la BBDD. Esta ordenación se hace de menor a mayor, dando preferencia a los recursos más pequeños.

La estructura recursiva es una tabla hash donde las claves son los recursos y los valores son un par formado por una lista de intercambios y otra estructura recursiva.

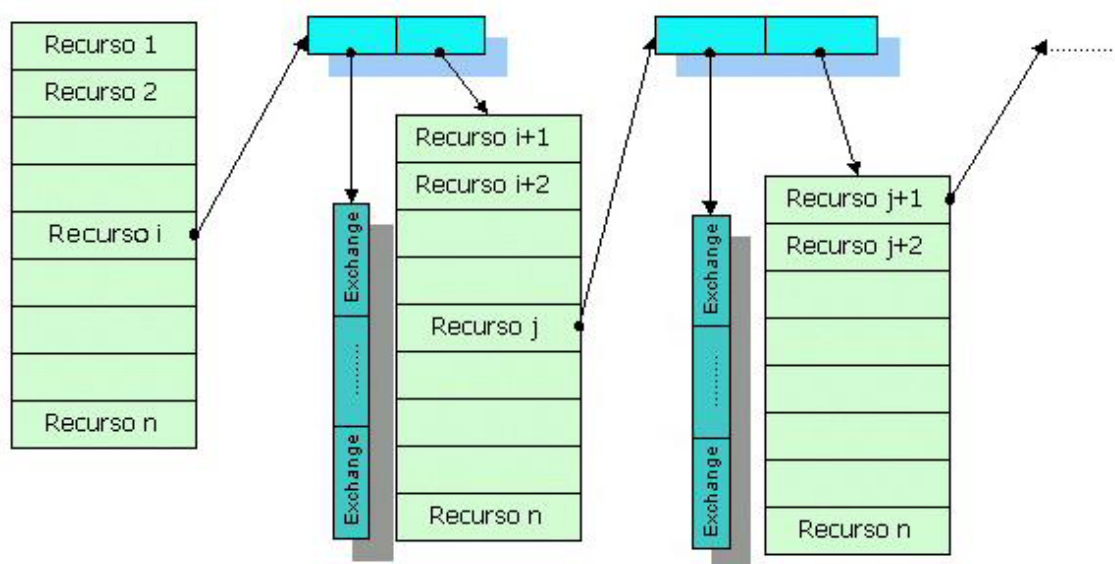


Fig. 6: Estructura para búsquedas

La idea es poner cada uno de los intercambios en todas las listas alcanzables desde la hash inicial con los recursos que se ofrecen. Es decir una intercambio que ofrezca a cambio de lo que quiere conseguir el recurso

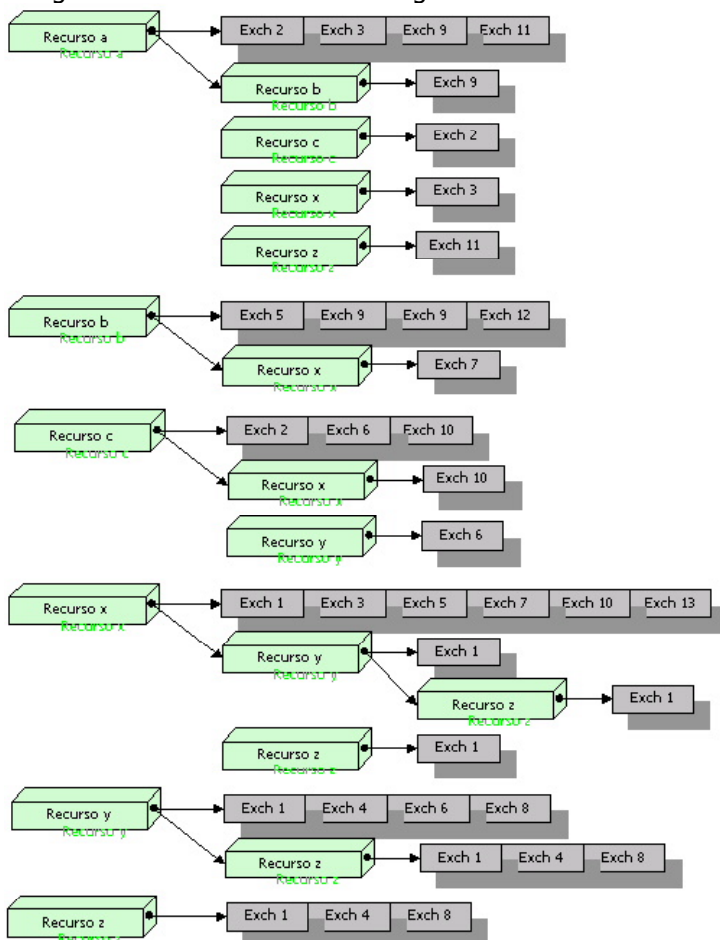


i y el recurso j deberá de aparecer en la listas que se ven en el dibujo, más la primera lista de la primera tabla hash correspondiente con el recurso j. Veamos un ejemplo.

Esta tabla representa a 13 intercambios con los recursos que desean obtener y los que ofrecen a cambio, y las cantidades.

Intercambio	Recursos que ofrece	Recursos que quiere obtener
Exch 1	$x + 2y + z$	$a + 2b$
Exch 2	$a + 2c$	$2y$
Exch 3	$x + a$	$x + z$
Exch 4	$y + z$	$2x$
Exch 5	$2b + x$	$2a + z$
Exch 6	$c + y$	$2a * x$
Exch 7	$x + 2b$	$2z + y$
Exch 8	$y + 2z$	$a + 2c$
Exch 9	$a + 2b$	$2y + x$
Exch 10	$2c + x$	$2y + z$
Exch 11	$2z + a$	$2b + c$
Exch 12	b	$2c + y$
Exch 13	x	$y + b$

Si tenemos en cuenta una ordenación de recursos como la siguiente, $a < b < c < x < y < z$ la estructura una vez generada debe de salir de la siguiente manera:



Con esta estructura se puede observar como buscar un intercambio que tenga el recurso a y el recurso b que está buscando el intercambio 1 es muy sencillo. Los intercambios 2,3,9 y 11 tienen el recurso a y además el intercambio 9 tiene el recurso b.

La opción sería buscar todos aquellos recursos que tienen al menos el recurso de menor peso, de forma que si consigues no consigues el recurso de menor peso no es necesario seguir pues aunque logres el resto siempre te faltará el menor.

En este caso el intercambio 1 empezará intentando negociar con el 9 pues es el que tiene más recursos de los que él necesita.

Si el intercambio 9 tiene la cantidad necesaria de recurso a y acepta lo que le ofrece el recurso 1 ya hemos logrado un intercambio virtual.

Ahora el 1 sólo necesita buscar lo que le falta del recurso b.

El proceso seguido podría ser el siguiente:

1. Cogemos el intercambio Exch 1:



- a. ¿Quién tiene el recurso a y el recurso b? -> Posibles candidatos: 9, 2, 3, 11 (El 9 va primero pues es el que además de darle a también le da b).
- b. Probar con el Exch 9: Los intercambios 1 y 9 realizan la transacción pues tanto el 9 como el 1 quedan plenamente satisfechos.
- c. Ahora hay que quitar de la estructura a los intercambios 1 y 9 y a todos los relacionados de ambos.
2. Cogemos el intercambio Exch2:
- a. ¿Quién tiene el recurso y? -> Posibles candidatos: (6,4,8)
- b. Probar con el Exch 6: No se puede hacer la transacción pues el 2 no es capaz de satisfacer al Exch 6.
- c. Probar con el Exch 4: No se puede hacer la transacción pues el 2 no es capaz de satisfacer al Exch 4.
- d. Probar con el Exch 8: Se realiza una transacción virtual pues aunque el Exch 8 queda satisfecho, el Exch 2 le falta un recurso y.
- i. ¿Quién tiene el recurso y? No se ha logrado completar la transacción
- e. Deshacer la transacción virtual
- f. Como no lo hemos conseguido dejamos al Exch 2 dentro de la estructura por si alguno de los exchanges que aún quedan desean negociar con el.

.....

Si suponemos que no hemos conseguido más transacciones ahora podemos empezar con las colaboraciones. Empezaremos con las de nivel 2 y en caso de que nos sigan faltando exchanges por conseguir probaremos con los niveles superiores.

1. Crear colaboración entre Exch 2 y Exch 3: Ahora la colaboración desea de conseguir los recursos $x + z + 2y$, y ofrece a cambio los recursos $2a + 2c + 2x$
 - a.
2. Crear colaboración Exch 2 y Exch 4.....

El máximo nivel que vamos a intentar está definido en la tabla System_Parameters de la BBDD.

3 Diagramas de clases

Las siguientes figuras muestran las distintas clases del sistema y sus relaciones.

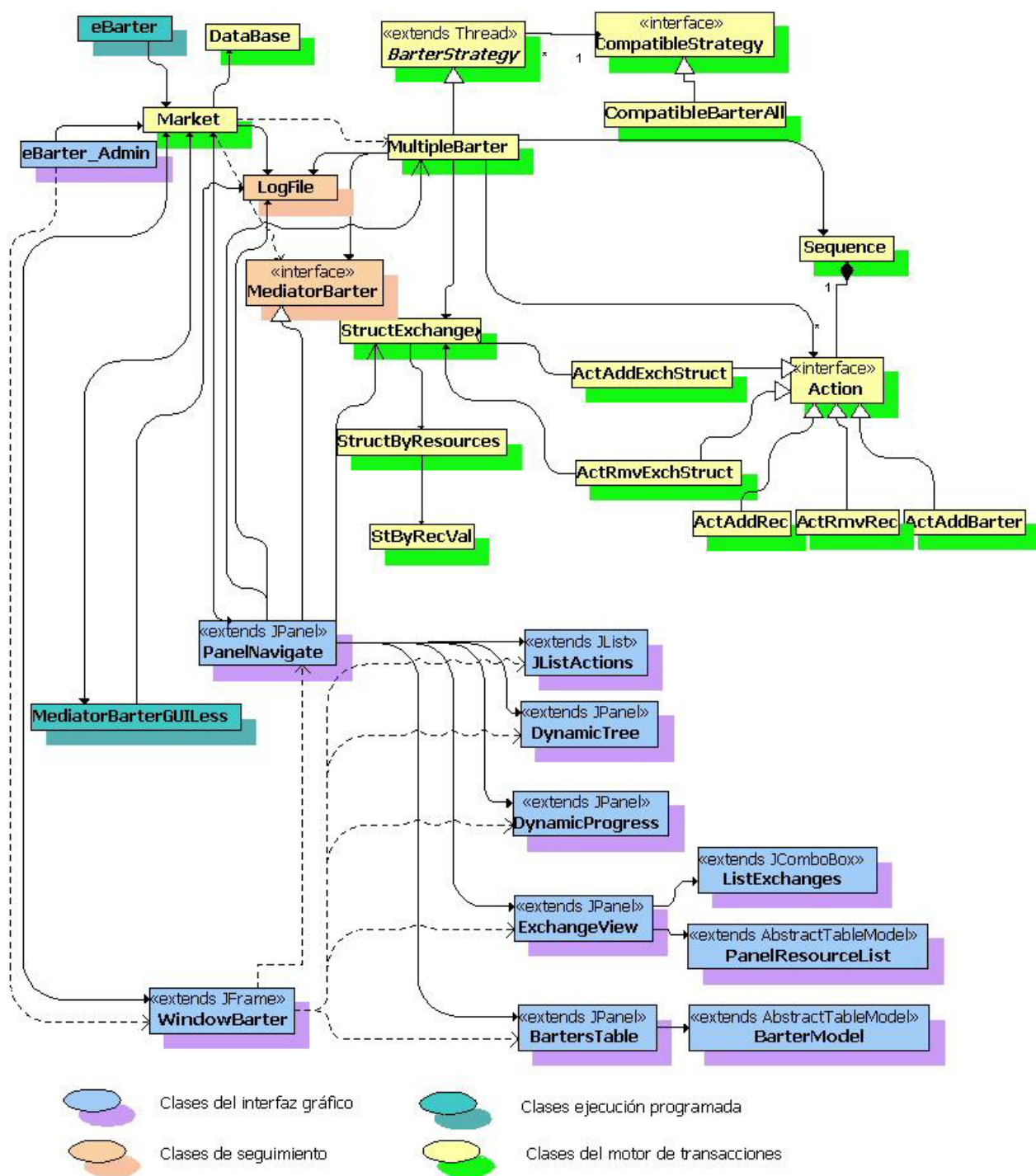


Fig. 7: Diagrama de clases de la aplicación

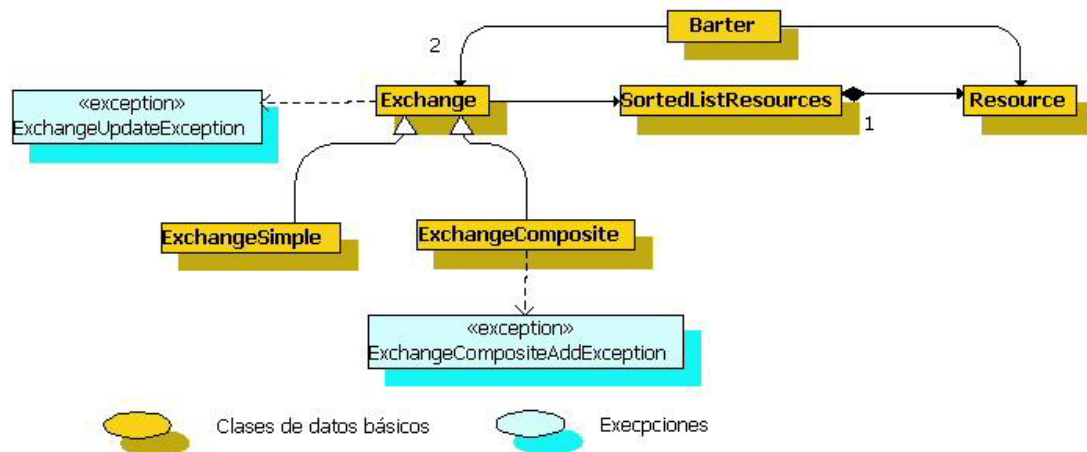
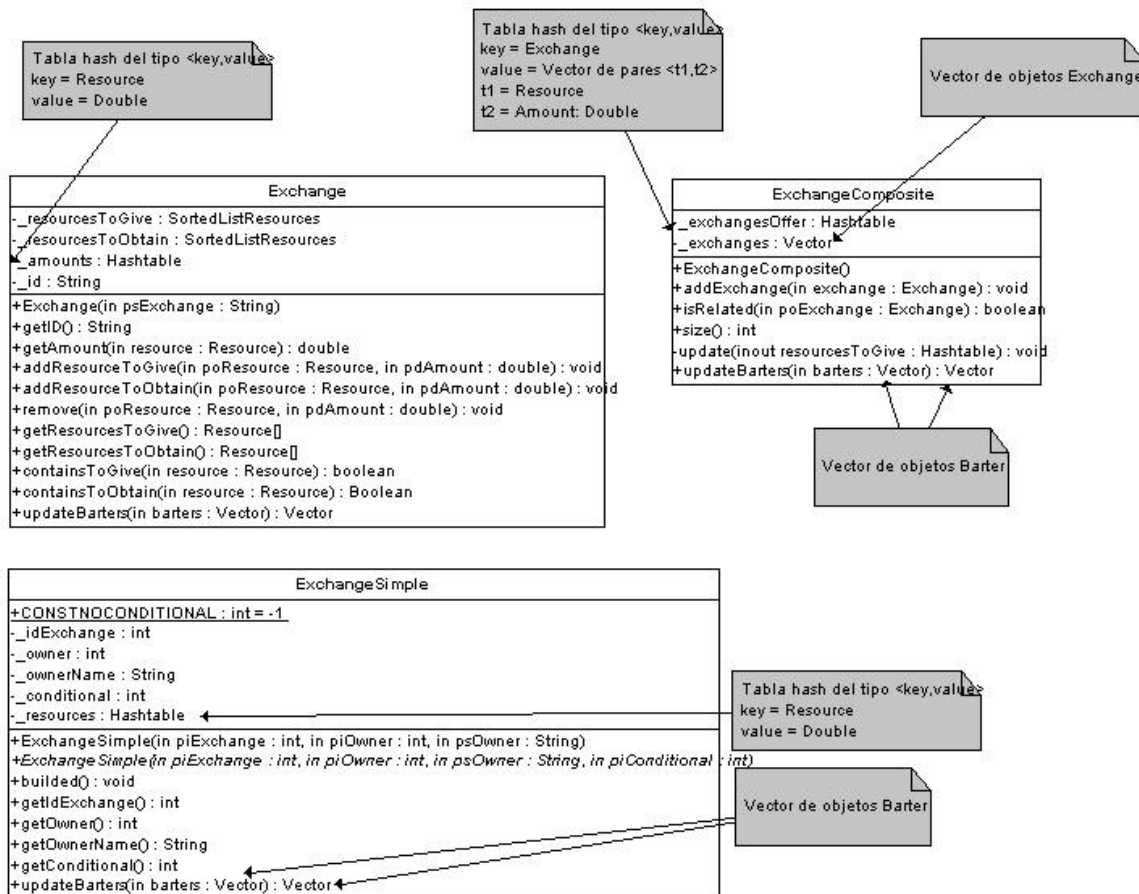


Fig. 8: Diagrama de clases para los datos básicos

Veamos ahora cada una de las clases más en detalle, atendiendo a los atributos que tienen y los servicios que ofrecen.



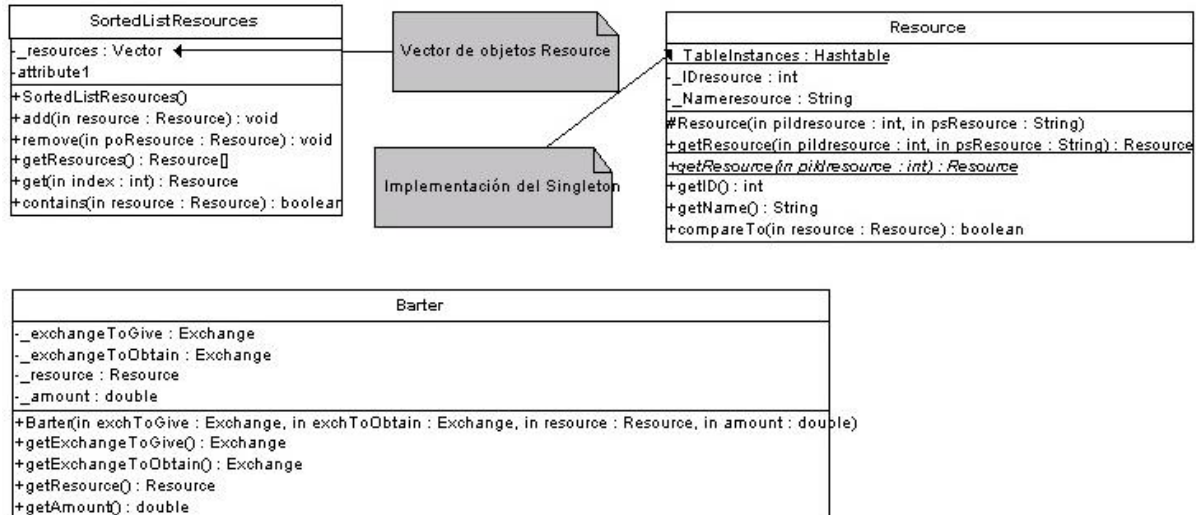
- Las clase Exchange es el tipo de datos que representa a los intercambios. Almacena información de los recursos que se desea conseguir y los que se está dispuesto a ofrecer.
 - ExchangeSimple representa a los intercambios de los usuarios. Almacena información sobre el usuario dueño del intercambio y el código condicional que tiene asignado.
 - ExchangeComposite representa a las colaboraciones entre exchanges. Almacena la información de los intercambios propietarios de los recursos que forman la colaboración.



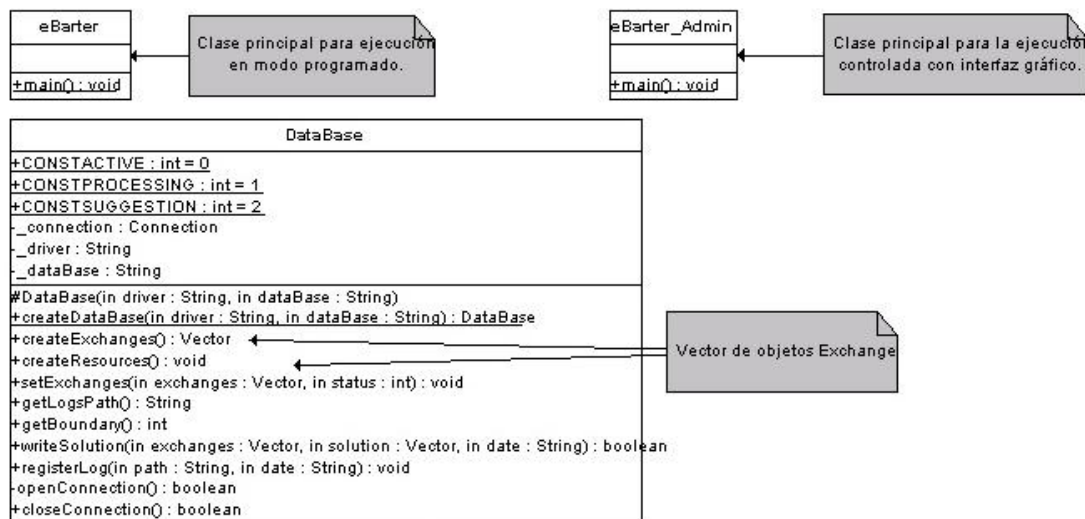
Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

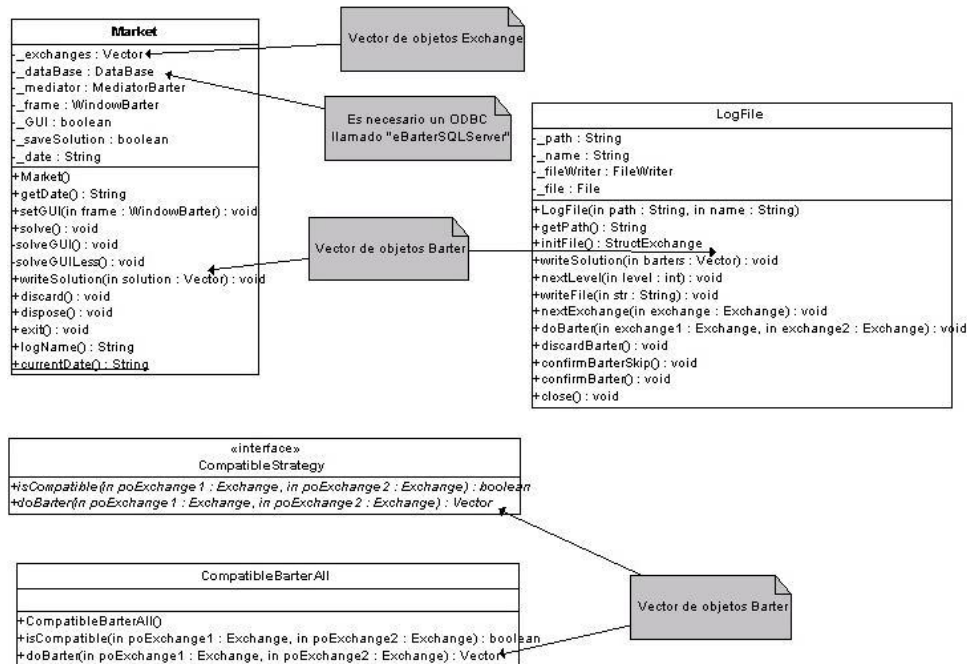
Título Documento : Memoria del proyecto e-Barter
Descripción : Explicación del funcionamiento interno del proceso de generación de transacciones.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 2: **MOTOR DE TRANSACCIONES** Pagina : 25/133



- La clase Barter representa a las transacciones: Mantiene información del intercambio origen, él de destino, el recurso a intercambiar y la cantidad. Varios objetos de esta clase formarán una transacción completa.
- La clase Resource representa a los recursos del sistema. Está implementado como singleton de forma que no creen varios objetos para representar al mismo recurso.
- La clase SortedListResources es una lista de recursos ordenada por los identificadores de éstos en la BBDD.



- La clase eBarter sólo contiene el método main para la ejecución programada
- La clase eBarter sólo contienen el método mail para la ejecución con interfaz gráfico.
- La clase DataBase es la única vía de comunicación de la aplicación con la BBDD. Crea los recursos y los intercambios que entraran en el proceso.



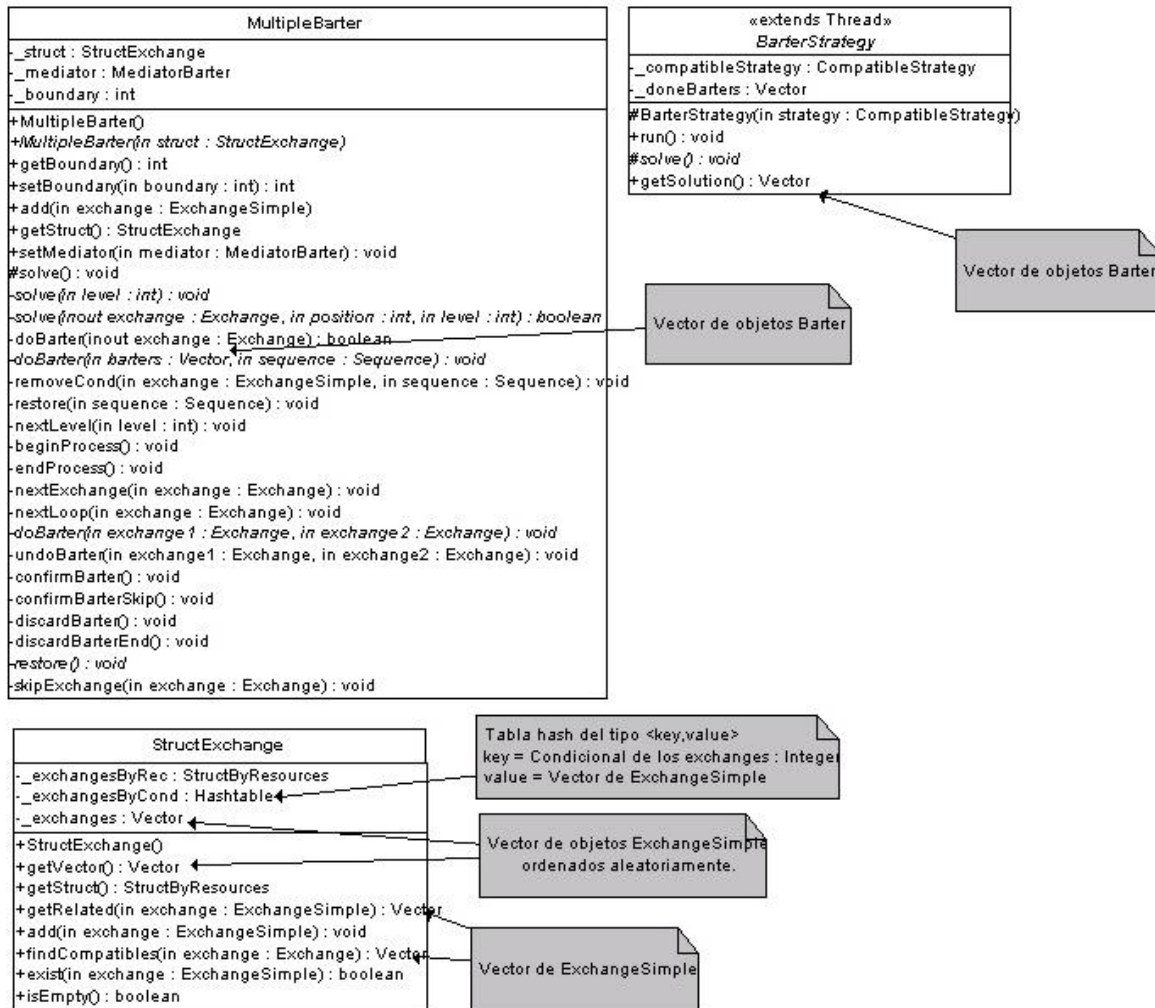
- La clase market es la clase que comunica e inicializa al resto de las clases de la aplicación. Sirve de puente entre la aplicación y la clase de acceso a la base de datos. También crea el fichero de log y el frame del interfaz gráfico.
- La clase CompatibleStrategy es una estándar que deben de cumplir las estrategias de compatibilidad, las cuales deben de decidir cuando dos intercambios pueden o no negociar y en que estado se quedan ambos después de la transacción.
- La clase CompatibleBarterAll es una implementación de una estrategia de compatibilidad en la cual dos intercambios sólo son compatibles si el primero de ellos queda totalmente satisfecho con la transacción.
- La clase LogFile sirve para guardar en un fichero de texto todo lo que ha pasado en la aplicación.



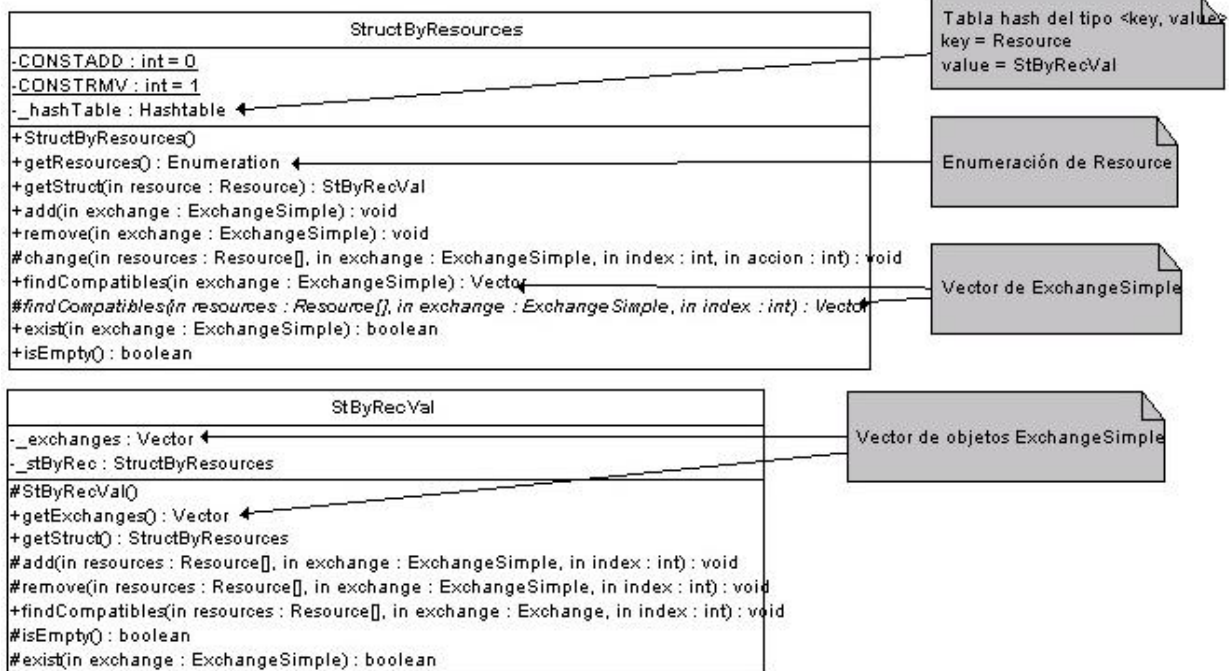
Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

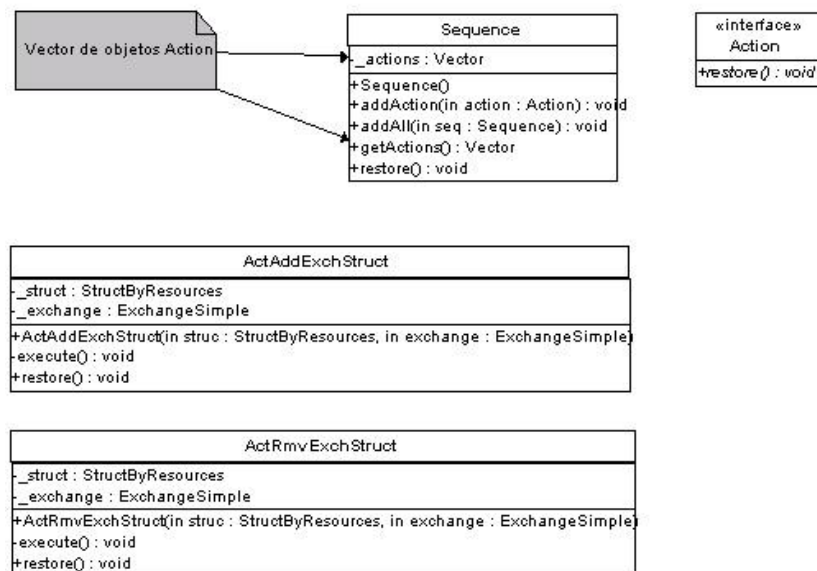
Título Documento : Memoria del proyecto e-Barter
Descripción : Explicación del funcionamiento interno del proceso de generación de transacciones.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 2: **MOTOR DE TRANSACCIONES** Pagina : 27/133



- La clase BarterStrategy es una clase abstracta que define los métodos que debe de implementar una estrategia de generación de transacciones. Además extiende a la clase Thread para poder realizar la búsqueda en un hilo de ejecución diferente al resto de la aplicación.
- MultipleBarter implementa una estrategia de búsqueda de transacciones concreta. Tiene una referencia a un objeto MediatorBarter al que le irá enviado mensajes sobre todo lo que suceda para realizar la auditoría. La estrategia se apoya en una estrategia de compatibilidad que dirá cuando dos intercambios pueden intercambiar entre sí y cual es el estado de cada uno tras la transacción.
- La clase StructExchange es la estructura sobre la que se apoya el algoritmo de búsqueda. Esta estructura está formada por un vector de intercambios, una tabla hash donde se guardan los intercambios relacionados y la estructura recursiva de la que hablamos anteriormente (StructByResources).



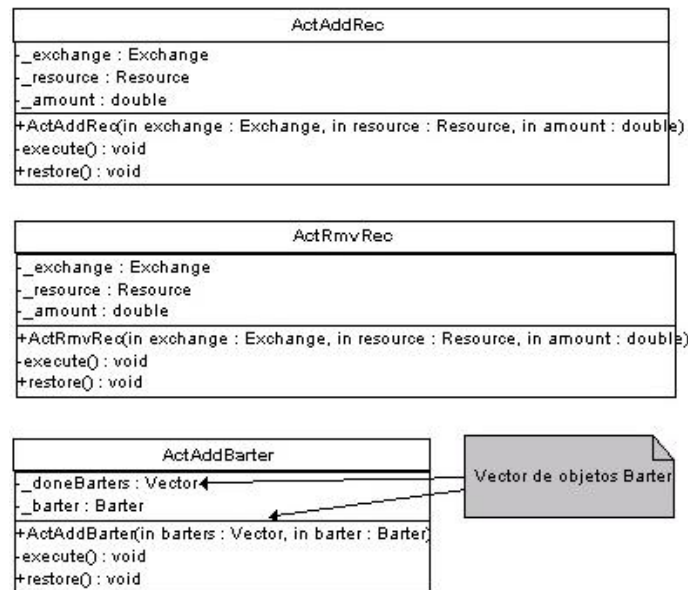
- La clase StructByResources representa la tabla hash recursiva que vimos anteriormente. Contiene métodos para añadir y eliminar intercambios de la estructura.
- La clase StByRecVal es la otra parte de la estructura recursiva, la que representa al campo Value de la tabla hash. Esta formada por un vector de intercambios y una tabla hash para completar la recursión.



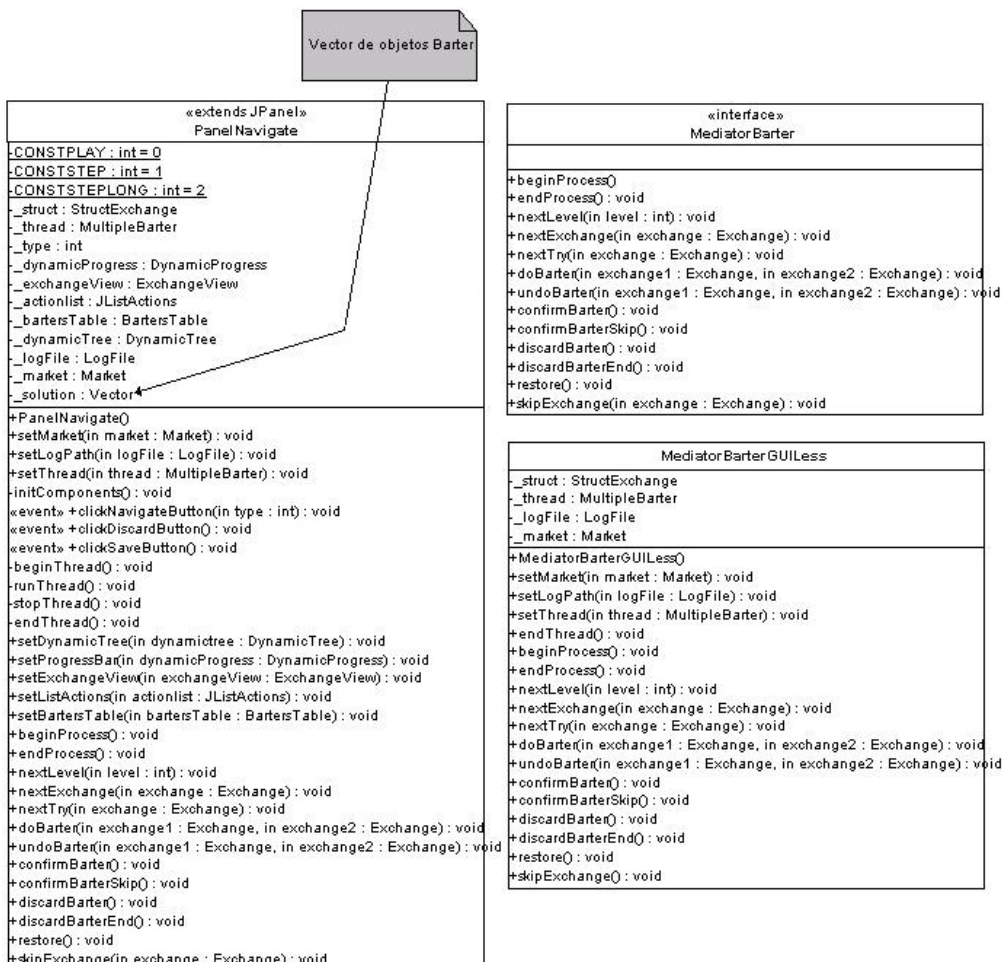
- La clase Sequence representa a una secuencia de acciones implementada para guardar todos los pasos que se hayan dado durante el proceso y posteriormente en caso de tener que hacer backtracking poder deshacer todas y cada una de las acciones hasta volver al estado inicial.
- La clase Action es un interfaz que deben de implementar todas las acciones que ocurren en el proceso. Estos métodos a implementar son el constructor y el método restore para devolver todo a su estado original.
- La clase ActAddExchStruct implementa la acción de incluir un intercambio en la estructura recursiva. El método restore debe de sacar el intercambio de la estructura.



- La clase ActRmvExchStruc implementa la acción contraria.



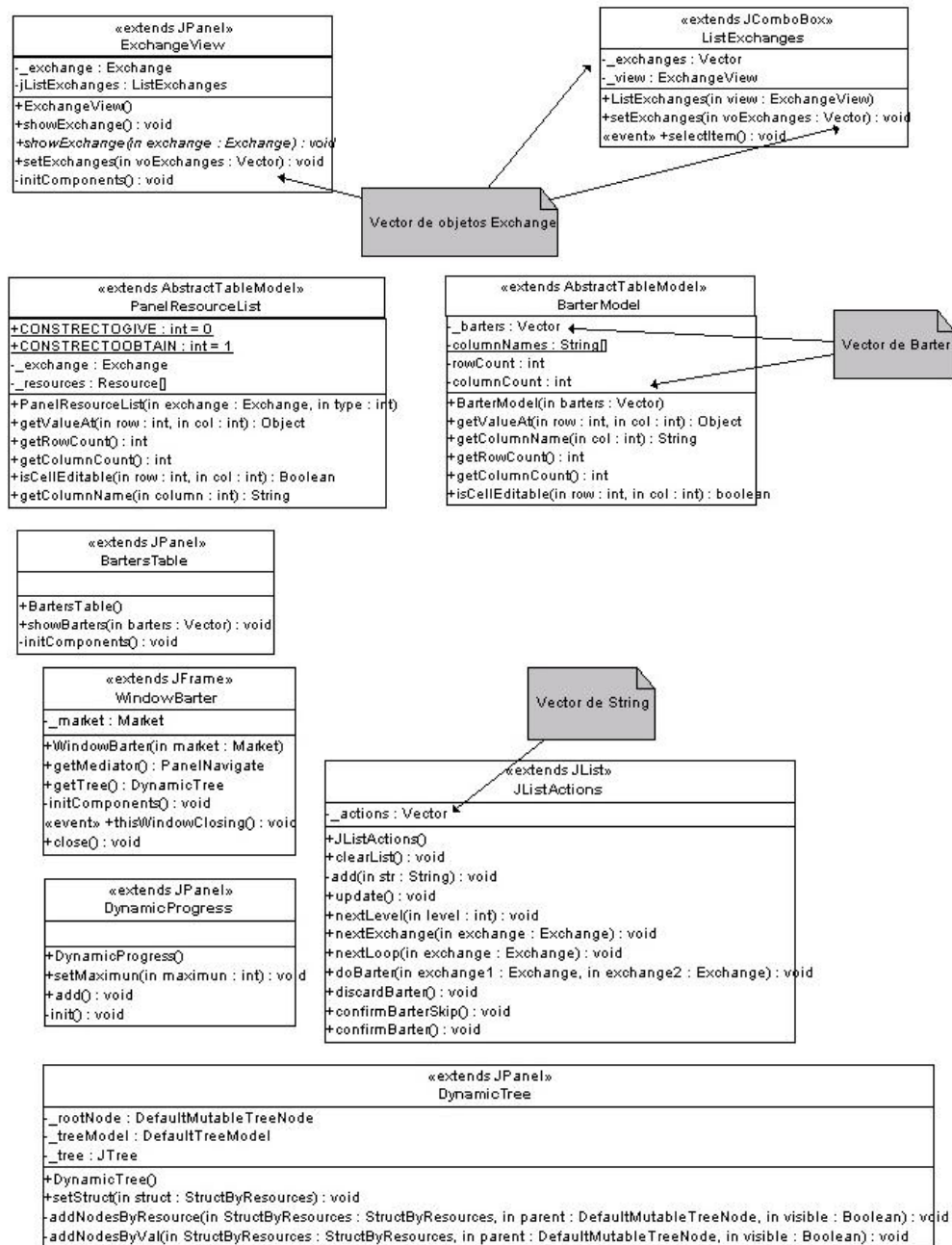
- La clase ActAddRec implementa la acción de añadir un recurso a un intercambio. El método restore debe de eliminar la misma cantidad del recurso.
- La clase ActRmvRec implementa la acción opuesta.
- La clase ActAddBarter implementa la acción de crear una transacción virtual.





- La clase MediatorBarter es un interfaz que deben de implementar todas las clases que deseen hacer un seguimiento de lo que sucede en motor de transacciones. Todos los métodos corresponden con mensajes que enviará la clase MultipleBarter a sus observadores.
- La clase MediatroBarterGUIless corresponde con el observador del modo pogramado que únicamente escribe los mensajes en un fichero de log. También dice al market que guarde los resultados.
- La clase PanelNavigate corresponde con el observador del modo GUI. Realiza un gran número de actualizaciones para cada mensaje que recibe. Es capaz de para la ejecución del proceso y reanudarla cuando el usuario desee.

El resto de clases sirven para implementar el interfaz gráfico del modo GUI:





4 Salida del motor de transacciones

El motor de transacciones recibe como entradas una serie de intercambios y sus relaciones. Todos estos intercambios que obtenemos como entrada no deberían ser modificados desde otras aplicaciones del sistema, por lo tanto se les pone el estado de PROCESSING. Tras aplicar todo el proceso que hemos descrito anteriormente obtenemos un conjunto de transacciones que satisfacen a algunos de estos intercambios de entrada. Ahora para los intercambios implicados en alguna de estas transacciones se les pone el estado EBARTER_SUGGESTION de modo que el usuario puede confirmar y descartar la transacción. Para todos los intercambios que tengan otro intercambio relacionado en estado EBARTER_SUGGESTION se les debe de poner en estado de DISCARD_BY_OR_RELATION. El resto de los intercambios de entrada deben de volver a su estado inicial de ACTIVE.

A parte de esto, la aplicación también es capaz de generar un completo archivo de Log cuya ubicación está definida en la base de datos y cuyo nombre de archivo se corresponde con la fecha y hora del momento que se creó.

Este fichero de Log tiene la siguiente estructura:

```
*****Exchanges List*****
#Lista con los intercambios que entran en el proceso. Están en el orden en que se van a procesar.

Exchange :27253
User :19851
Related :-1
Resources to give :ORANGES(3.7)      TUNA(6.6)
Resources to obtain :APPLES(5.8)      MELONS(1.6)

Exchange :12375
User :23522
Related :-1
Resources to give :COKE(3.6)
Resources to obtain :APPLES(5.4)
...
*****Proccess*****
# Seguimiento del algoritmo paso a paso.

Level: 1
EXCHANGE-> 27253
Try to-> 3742
Try to-> 18576
Try to-> 21572
Try to-> 28792
EXCHANGE-> 1924
Try to-> 6122
Try to-> 15839
Try to-> 20349
Try to-> 21003
Try to-> 25502
Try to-> 29045
Barter between-> 29045 and 1924
Barter succeeded
...

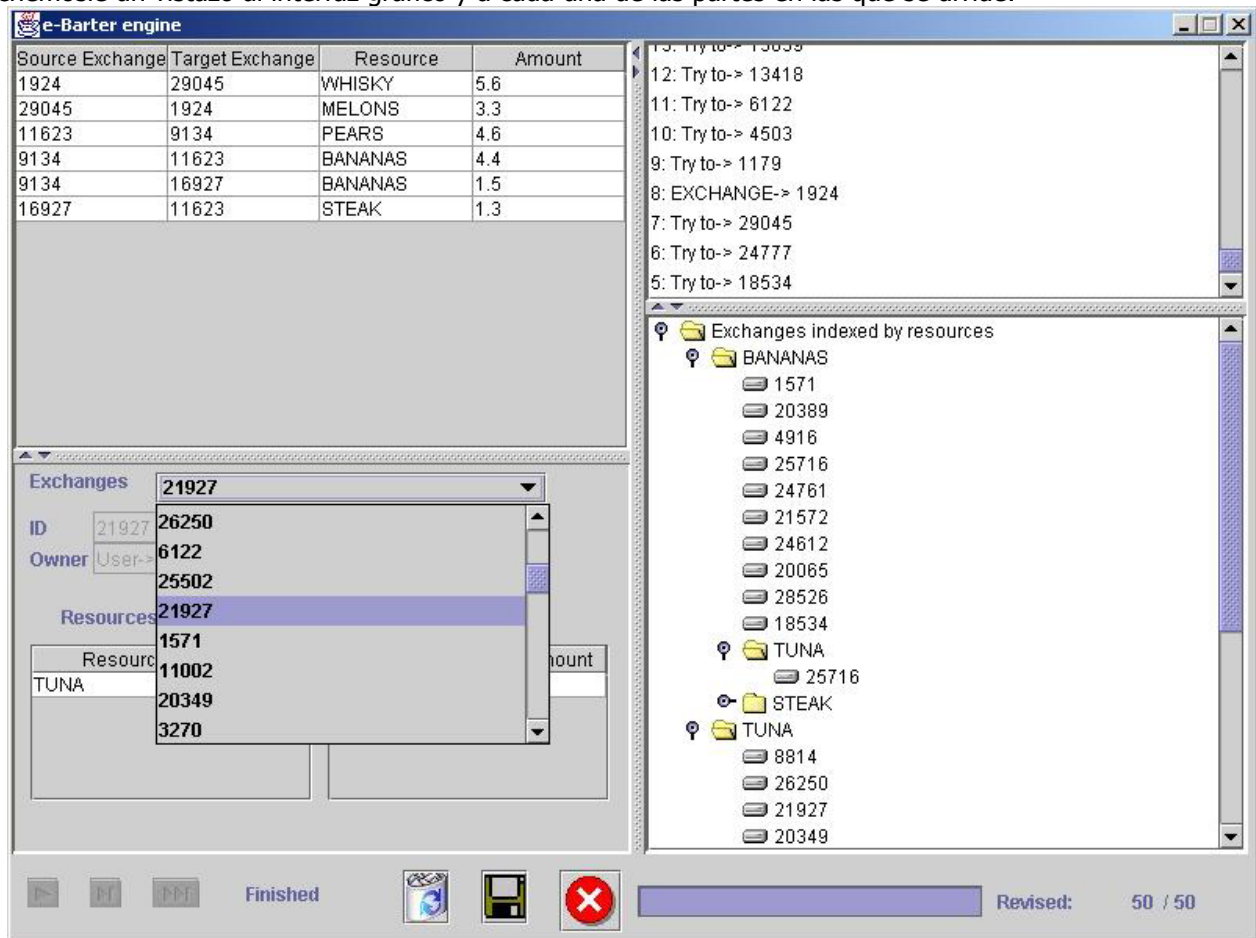
Level: 2
...
*****Solution*****
# Transacciones logradas
Exchange source :1924
Exchange target :29045
Resource :WHISKY
Amount :5.6
....
*****THE END*****
# Indica si se grabó la solución o hubo algún problema
Save solution...
```

La última línea del registro de Log sólo puede tomar 3 valores o no existir:




- No existe: Se produjo una salida busca de la aplicación y el proceso no se completó. Todos los intercambios en la base de datos tendrán el estado PROCESING y será tarea del administrador cambiar esos estados.
- Save solution...: El proceso se ha completado, se han grabado las transacciones en la base de datos y se han actualizado los estados de los intercambios.
- Error to save solution...: Tras completarse el proceso no se han podido grabar las transacciones logradas en la base de datos. Todos los intercambios han vuelto al estado ACTIVE. Generalmente el error se deberá a un error en la conexión a la base de datos o que hay basura en la tabla EBARTER_TRANSACTION. Sería recomendable que el administrador investigase los motivos del fallo.
- Discard solution...: Esto se produce cuando en el modo GUI el usuario decide no grabar la solución que le ofrece la aplicación. Todos los intercambios han vuelto al estado ACTIVE.


Echémosle un vistazo al interfaz gráfico y a cada una de las partes en las que se divide:





La pantalla se divide en 6 partes bien diferenciadas:


- La parte inferior derecha muestra una barra de progreso. Esta barra tiene como límite el número de intercambios que entran en juego multiplicado por el límite de colaboraciones que podemos establecer. (En el ejemplo hay 25 intercambios y pueden colaborar 2 intercambios como máximo).
- En la parte inferior izquierda podemos ver una barra de navegación con tres botones y una etiqueta.


El botón  permite la ejecución ininterrumpida del motor.

El botón  realiza una ejecución paso a paso.

El botón  permite navegar hasta el siguiente intercambio a comprobar.

El botón  permite descartar la solución ofrecida y volver a ejecutar el algoritmo.

El botón  graba las transacciones resultantes y acaba la ejecución.

El botón  termina con la ejecución en cualquier momento.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter
Descripción : Explicación del funcionamiento interno del proceso de generación de transacciones.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 2: **MOTOR DE TRANSACCIONES** Pagina : 33/133

- En la parte media derecha hay una representación de la estructura recursiva para almacenar los intercambios. Los nodos desplegados son recursos representados por su name, y las hojas son intercambios representados por su identificador de la base de datos.
- La parte media izquierda permite ver el estado actual de los intercambios de modo que se pueda observar como varían los recursos que tiene.
- La parte superior derecha muestra la lista de acciones que se han realizado. Es la misma información que se encuentra en el fichero de Log.
- La parte superior izquierda de la pantalla muestra las transacciones encontradas ya aplanadas¹.

¹ Cuando una colaboración encuentra una transacción virtual se establece entre la colaboración y un intercambio. Cuando la transacción pasa de virtual a real esta transacción se aplanar, eliminando las referencias a la colaboración y convirtiéndolas a intercambios simples concretos pertenecientes a la colaboración.



5 Estadísticas

Se han realizado diversas pruebas de carga sobre el algoritmo de generación de transacciones. Somos conscientes que el número de pruebas realizadas y el tamaño de la muestra no es lo suficientemente amplio como para sacar unos resultados definitivos, pero si son bastantes representativos principalmente de su aleatoriedad.

Todas estas pruebas han sido realizadas sobre un Pentium II con 350Mz y 320 MB de memoria. Todas las pruebas se han realizado en modo programado pues la incorporación de un interfaz gráfico dispara el consumo de memoria y en consecuencia el tiempo de respuesta.

Para las pruebas se crearon intercambios generados aleatoriamente mediante un programa desarrollado a tal efecto. Con dicho programa se generaron 50 usuarios aleatoriamente y se crearon diferente número de intercambios asociados a los usuarios. Estos intercambios tenían a lo sumo cuatro recursos en total (entre los que dan y los que reciben) repartidos de forma aleatoria con al menos un recurso para ofrecer y otro para conseguir. Todas las colaboraciones también se generaron aleatoriamente entre los intercambios pertenecientes a un mismo usuario.

Hay que tener en cuenta que la complejidad no es ni mucho menos lineal y por lo tanto la incorporación de unos pocos intercambios al proceso puede multiplicar el tiempo de respuesta. Esto es mucho más acusado cuando introducimos las colaboraciones.

Se han tomado datos para 200, 300 y 500 intercambios para niveles de intercambio 1 y 3 obteniéndose resultados bastante satisfactorios.

Hay que destacar que la ordenación aleatoria de los intercambios previa a la ejecución del algoritmo hace que varíen mucho algunas de las pruebas tomadas sobre los mismo intercambios y sobre el mismo nivel de colaboración.

Pruebas sobre 300 intercambios con nivel 1 de colaboración

Tiempo de respuesta	Número de intercambios satisfechos	Número de intercambios descartados por las relaciones	Número de MB del fichero de Log.
15"	115	11	2
6"	112	15	0.3
10"	107	15	0.9

En esta caso los tiempos son demasiado pequeños como para poder sacar conclusiones.

Pruebas sobre 500 intercambios con nivel 1 de colaboración

Tiempo de respuesta	Número de intercambios satisfechos	Número de intercambios descartados por las relaciones	Número de MB del fichero de Log.
27"	195	15	3
13"	205	21	0.5
9"	190	26	0.5

Aquí empezamos a ver como bajo las misma condiciones una de las ejecuciones ha llegado a tardar hasta 3 veces más que otra, y sin embargo ha encontrado menos intercambios satisfechos.



Pruebas sobre 200 intercambios con nivel 3 de colaboración

Tiempo de respuesta	Número de intercambios satisfechos	Número de intercambios descartados por las relaciones	Número de MB del fichero de Log.
55"	101	12	15
35"	105	13	15
58"	110	10	16
1'	105	12	18
20"	104	15	6.7

Ya podemos apreciar como hay grandes oscilaciones tanto en tiempo de respuesta (20" – 1') como en el número de pruebas realizadas (6.7MB – 18MB). Sin embargo no hay tanta diferencia entre el número de éxitos logrados.

Pruebas sobre 300 intercambios con nivel 3 de colaboración

Tiempo de respuesta	Número de intercambios satisfechos	Número de intercambios descartados por las relaciones	Número de MB del fichero de Log.
1' 35"	162	15	30
5'	165	12	80
1' 15"	153	19	19
2' 15"	149	11	11

En esta caso se puede ver claramente grandes diferencias. Tiempos de respuesta que oscilan entre 1' 35" y 5', y fichero de Log con diferencias de 70MB, con una diferencia mínima entre el número de éxitos.

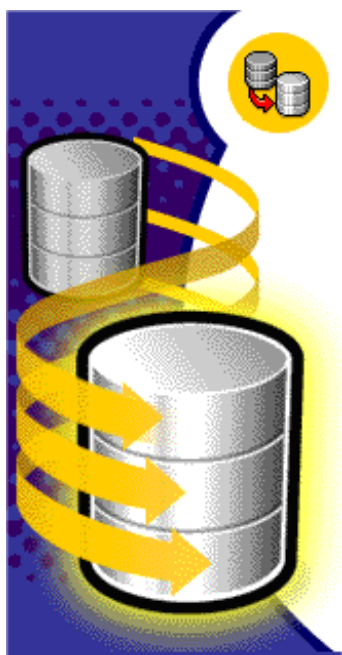
También se puede ver la utilidad de introducir colaboraciones. Para el mismo número de intercambios hemos obtenido alrededor de 40 intercambios satisfechos más.

Al realizar pruebas para 500 intercambios de nivel 3 no hemos conseguido tener ninguna respuesta.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.
Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter
Descripción : Diseño de la base de datos y código Transac-SQL.
Fecha. Creación : 01 Julio 2002
Sección 3: **BASE DE DATOS**
Versión 1.0.0
Pagina : 36/133



SECCION 3. : BASE DE DATOS



1 Introducción. Objetivos y Alcance del documento

En este documento se presenta toda la información relacionada con la base de datos que proporciona el soporte necesario para el almacenamiento de la información requerida por el sistema e-barter para su funcionamiento. Esto incluye además de las tablas un completo soporte de Código embebido en la propia base de datos a través de un conjunto de procedimientos de almacenado, y desencadenadores (triggers)

En el documento se incluye los diagramas correspondientes al diseño de la base de datos, pasando posteriormente a describir cada una de las tablas y campos incluidos en la misma, finalizando con la descripción de los procedimientos de almacenado utilizados.

El diseño final del sistema, la estructura y los tipos de datos aquí presentados se basan en la implementación física sobre un RDBMS SQL Server 2000 (Personal Edition). No obstante, cabe mencionar que las pruebas y test sobre los prototipos iniciales fueron realizadas sobre un sistema bastante menos complejo como es ACCESS 2000.

El cambio del gestor de base de datos desde el prototipo a la versión final es debido a la inclusión de Código mencionada anteriormente. Esto nos permite que las operaciones de mantenimiento de datos como son el mantenimiento de los recursos del usuario (los que tiene y pone en juego) la sincronización y la evolución de los estados de los intercambios registrados en el sistema, se encuentren centralizados en la propia base de datos y sean utilizados y compartidos desde el cliente WEB, la aplicación de administración y el motor de intercambios. Además de flexibilizar el mantenimiento permitiendo que un cambio en la estructura y definición de las tablas solo requiera la modificación de dichos procedimiento.

En el diseño e implementación se ha intentado facilitar posibles migraciones a otros sistemas (ejemplo ORACLE) evitando el uso de tipos y/o estructuras particulares (tipos Autoincrementales vs. Secuencias, etc).

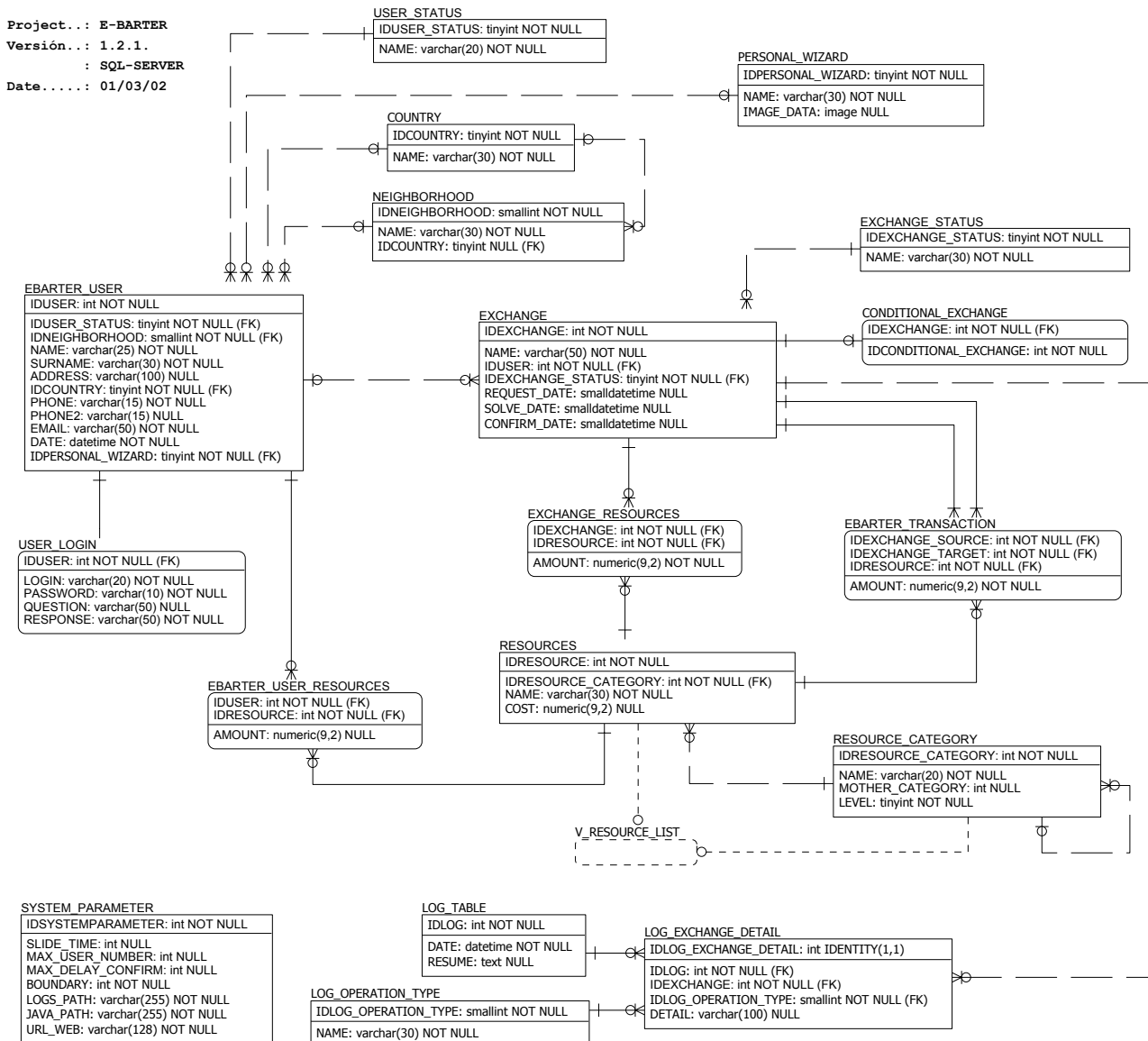
Para el diseño e implementación de la base de datos se han utilizado las siguientes herramientas.

1. Er-win (Herramienta CASE para el diseño y análisis de datos)
2. MS-Access 2000.
3. SQL_server 2000 / Personal Edition



2 Diagrama Base de Datos.

A continuación se presenta el diagrama con el modelo de tablas para el sistema de intercambio de recursos (e-barter).





3 Catalogo-Referencia de Tablas.

Table Name	Table Comment
CONDITIONAL_EXCHANGE	Tabla complementaria que almacena los Códigos de intercambios relacionados a través de una relación condicional de concesión de los recursos. Permite registrar y establecer relaciones entre intercambios con condiciones OR de tal forma que solo uno de los intercambios se realice.
COUNTRY	Tabla de referencia que contiene los diferentes países. Utilizado por la tabla EBARTER_USER para el registro de los datos personales de cada uno de los usuarios.
EBARTER_TRANSACTION	Table to register the complete transaction making in the e-barter system.
EBARTER_USER	Tabla que contiene los diferentes usuarios que están registrados en el sistema ebarter. Contiene sus datos personales y de gestión interna requeridos. Cada usuario se corresponde posteriormente con un agente en el procesamiento del motor ebarter.
EBARTER_USER_RESOURCES	Tabla que relaciona a cada uno de los usuarios registrados con los recursos de que disponen. Los datos no son requeridos para la gestión de los intercambios a través del motor eBarter, pero si es usada en la interfaz web para realizar un filtro en el alta de un nuevo intercambio.
EXCHANGE	Tabla que contiene todas las peticiones de intercambio realizadas en el sistema eBarter. Los elementos eliminados por los usuarios se dan de baja a través del establecimiento de un estado especial (ELIMINADO) permitiendo la no visualización de los mismos, pero no se elimina físicamente del registro.
EXCHANGE_RESOURCES	Tabla que relaciona cada uno de los intercambios con los recursos incluidos en el sistema e incluidos en dichas peticiones, almacenando la cantidad de los mismos que se esta dispuesto a dar (signo positivo) o que se quiere a cambio (signo negativo)
EXCHANGE_STATUS	Tabla de referencia que contiene los diferentes posibles estados en los que se pueden encontrar los intercambios del sistema. Permite una clasificación flexible de los mismos.
LOG_EXCHANGE_DETAIL	Tabla de LOG que almacena de forma mas detallada (por intercambios gestionados), las diferentes acciones u operaciones (registradas en la tabla LOG_OPERATION_TYPE) llevadas a cabo en cada ejecución del sistema. (registrados en la tabla LOG_TABLE). Permite realizar un seguimiento completo de la gestión y procesamiento que se ha realizado para cada uno de los intercambios que han sido evaluados por el sistema.
LOG_OPERATION_TYPE	Tabla de referencia que contiene cada una de las acciones u operaciones individuales que se desean registrar en las tablas de log durante la ejecución del motor de búsqueda de intercambios compatibles.
LOG_TABLE	Tabla que almacena la información general de log para cada una de las ejecuciones del motor de búsqueda de intercambios compatibles.
NEIGHBORHOOD	Tabla de referencia que registra los diferentes vecindarios o distritos que son asociados a los usuarios del sistema
PERSONAL_WIZARD	Tabla de referencia que almacena la información de los diferentes asistentes disponibles para la personalización de la interfaz web por parte de los usuarios.
RESOURCE_CATEGORY	Tabla de referencia que contiene las diferentes categorías en las que podemos clasificar los recursos y productos gestionados en el sistema. Permite el anidamiento y referencias recursivas para poder establecer una jerarquía de clases y subclases entre las diferentes categorías.
RESOURCES	Tabla con todos los diferentes recursos dados de alta en el sistema e_barter. Cada uno de ellos se encuentra englobado dentro de una categoría para facilitar su clasificación y posterior selección desde las interfaces de aplicación. Solo es posible realizar peticiones de intercambios sobre los objetos contenidos en esta tabla. El mantenimiento de esta tabla se realiza solo por el usuario administrador.
SYSTEM_PARAMETER	Esta tabla contiene los diferentes parámetros de configuración y limitaciones del sistema. Permite regular de forma simple diferentes aspectos del funcionamiento del sistema.
USER_LOGIN	Tabla que contiene los datos de identificación para el acceso vía cliente web para los diferentes usuarios registrados. Relacionada con la tabla EBARTER_USER en una relación 1:1. Los datos se almacenan por separado por cuestiones de seguridad.
USER_STATUS	Tabla de referencia que contiene los diferentes posibles estados en los que se pueden encontrar los usuarios registrados en el sistema e_barter. Permite tener una calificación entre usuarios activos, dados de baja, etc.
V_RESOURCE_LIST	Vista que nos proporciona los 100 primeros recursos casados con sus categorías y en un formato que facilita la gestión y visualización de los mismos por parte del cliente web.



4 Descripción física de Tablas

Table Name: CONDITIONAL_EXCHANGE

Table Comment: Tabla complementaria que almacena los códigos de intercambios relacionados a través de una relación condicional de concesión de los recursos. Permite registrar y establecer relaciones entre intercambios con condiciones OR de tal forma que solo uno de los intercambios se realice.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDEXCHANGE	int	NOT NULL	P.K. Código numérico que identifica a cada uno de los intercambios. Proviene de la relación con la tabla EXCHANGE.	Yes	Yes
IDCONDITIONAL_EXCHANGE	int	NOT NULL	Código numérico común a todos los intercambios que estén relacionados a través de una ejecución condicional.	No	No

Table Name: COUNTRY

Table Comment: Tabla de referencia que contiene los diferentes países. Utilizado por la tabla EBARTER_USER para el registro de los datos personales de cada uno de los usuarios.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDCOUNTRY	tinyint	NOT NULL	Nombre de los diferentes Países a los que se puede asociar un usuario	Yes	No
NAME	varchar(30)	NOT NULL		No	No

Table Name: EBARTER_TRANSACTION

Table Comment: Table to register the complete transaction making in the e-barter system.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDEXCHANGE_TARGET	int	NOT NULL	Código numérico que identifica al intercambio de destino (el que recibe el recurso) en la operación propuesta por el sistema. Proviene de la relación con la tabla EXCHANGE	Yes	Yes
IDRESOURCE	int	NOT NULL	Código del recurso referenciado en la transacción. Proviene de la relación con la tabla RESOURCE.	Yes	Yes
AMOUNT	numeric(9,2)	NOT NULL	Cantidad cedida por el intercambio SOURCE al intercambio TARGET del recurso IDRESOURCE.	No	No
IDEXCHANGE_SOURCE	int	NOT NULL	Código numérico que identifica al intercambio de origen (el que da un recurso) en la operación propuesta por el sistema. Proviene de la relación con la tabla EXCHANGE	Yes	Yes

Table Name: EBARTER_USER

Table Comment: Tabla que contiene los diferentes usuarios que están registrados en el sistema e_barter. Contiene sus datos personales y de gestión interna requeridos. Cada usuario se corresponde posteriormente con un agente en el procesamiento del motor ebarter.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDPERSONAL_WIZARD	tinyint	NOT NULL	Código numérico de identificación del asistente seleccionado por el usuario en la interfaz web. F.K. que referencia a la tabla PERSONAL_WIZARD.	No	Yes
IDUSER	int	NOT NULL	P.K. de la tabla. Número único que identifica a cada uno de los usuarios del sistema ebarter.	Yes	No
IDNEIGHBORHOOD	smallint	NOT NULL	F.K. identificador numérico que identifica el vecindario o distrito al que está asociado cada usuario. Referencia a la tabla NEIGHBORHOOD.	No	Yes
IDUSER_STATUS	tinyint	NOT NULL	F.K. identificador del estado en el que se encuentra un usuario. Referencia a la tabla USER_STATUS	No	Yes
NAME	varchar(25)	NOT NULL	Nombre del usuario	No	No
SURNAME	varchar(30)	NOT NULL	Apellidos del usuario	No	No
ADDRESS	varchar(100)	NULL	Dirección del usuario	No	No
IDCOUNTRY	tinyint	NOT NULL	País.	No	Yes
PHONE	varchar(15)	NOT NULL	Teléfono principal del usuario	No	No
EMAIL	varchar(50)	NOT NULL	e-mail del usuario. Utilizado para mandar información acerca del sistema e_barter, como por ejemplo la lista de intercambios propuestos.	No	No
PHONE2	varchar(15)	NULL	Número de teléfono alternativo	No	No
DATE	datetime	NOT NULL	Fecha de alta del usuario en el sistema	No	No

Table Name: EBARTER_USER_RESOURCES



Table Comment: Tabla que relaciona a cada uno de los usuarios registrados con los recursos de que disponen. Los datos no son requeridos para la gestión de los intercambios a través del motor eBarter, pero si es usada en la interfaz web para realizar un filtro en el alta de un nuevo intercambio.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDUSER	int	NOT NULL	Valor participe en la clave de la tabla. identificador único para cada usuario del sistema. Proviene de la relación con la tabla EBARTER_USER	Yes	Yes
IDRESOURCE	int	NOT NULL	Valor participe en la clave de la tabla. Código numérico que identifica a cada uno de los intercambios. Proviene de la relación con la tabla EXCHANGE.	Yes	Yes
AMOUNT	numeric(9,2)	NULL	Cantidad del recurso que tiene el usuario.	No	No

Table Name: EXCHANGE

Table Comment: Tabla que contiene todas las peticiones de intercambio realizadas en el sistema eBarter. Los elementos eliminados por los usuarios se dan de baja a través del establecimiento de un estado especial (ELIMINADO) permitiendo la no visualización de los mismos, pero no se elimina físicamente del registro.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDEXCHANGE	int	NOT NULL	P.K. identificador de los registro de intercambio incluidos en la tabla. Código numérico único que identifica a cada uno de los intercambios incluidos en el sistema ebarter.	Yes	No
IDUSER	int	NOT NULL	identificador único del usuario al que pertenece el intercambio. F.K. que hace referencia a la tabla EBARTER_USER	No	Yes
NAME	varchar(50)	NOT NULL	Nombre descriptivo del intercambio empleado por los usuarios para una identificación mas cómoda.	No	No
IDEXCHANGE_STATUS	tinyint	NOT NULL	Código numérico que indica el estado en el que se encuentra el intercambio. FK. que referencia a la tabla EXCHANGE_STATUS	No	Yes
REQUEST_DATE	smalldatetime	NULL	Fecha de entrada o petición del intercambio en el sistema.	No	No
SOLVE_DATE	smalldatetime	NULL	Fecha de procesamiento y resolución por parte del gestor ebarter.	No	No
CONFIRM_DATE	smalldatetime	NULL	Fecha de confirmación del intercambio por parte del usuario. El tiempo de confirmación puede ajustarse a través de la diferencia entre el campo SOLVE_DATE y CONFIRM_DATE según lo indicado en la tabla de control y ajuste del sistema	No	No

Table Name: EXCHANGE_RESOURCES

Table Comment: Tabla que relaciona cada uno de los intercambios con los recursos incluidos en el sistema e incluidos en dichas peticiones, almacenando la cantidad de los mismos que se esta dispuesto a dar (signo positivo) o que se quiere a cambio (signo negativo)

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDEXCHANGE	int	NOT NULL	Código numérico único que identifica a cada uno de los intercambios incluidos en el sistema ebarter. Proviene de la relación con la tabla EXCHANGE	Yes	Yes
IDRESOURCE	int	NOT NULL	Código numérico que identifica a cada uno de los recursos incluidos en el sistema. Proviene de la relación con la tabla RESOURCE	Yes	Yes
AMOUNT	numeric(9,2)	NOT NULL	Cantidad del recurso que se esta dispuesta a dar (+) o que se quiere obtener (-) en el intercambio referenciado en el registro. El sentido depende del signo de la cantidad. (Dar=positivo / Obtener=Negativo)	No	No

Table Name: EXCHANGE_STATUS

Table Comment: Tabla de referencia que contiene los diferentes posibles estados en los que se pueden encontrar los intercambios del sistema. Permite una clasificación flexible de los mismos.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDEXCHANGE_STATUS	tinyint	NOT NULL	Primary Key de la tabla. identificador numérico para los diferentes estados en los que podemos encontrar los intercambios incluidos en el sistema.	Yes	No
NAME	varchar(30)	NOT NULL	Nombre del estado.	No	No

Table Name: LOG_EXCHANGE_DETAIL

Table Comment: Tabla de LOG que almacena de forma mas detallada (por intercambios gestionados), las diferentes acciones u operaciones (registradas en la tabla LOG_OPERATION_TYPE) llevadas a cabo en cada ejecución del sistema. (registrados en la tabla LOG_TABLE). Permite realizar un seguimiento completo de la gestión y procesamiento que se ha realizado para cada uno de los intercambios que han sido evaluados por el sistema.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Proyecto S.I. / e-Barter. (Año 2002/03)

Sección 3: BASE DE DATOS

Página : 42/133

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDLOG_EXCHANGE_DETAIL	int	IDENTITY	P.K. identificador de los registros de la tabla	Yes	No
IDLOG	int	NOT NULL	F.K. identificador que referencia al log de cada una de la ejecuciones realizadas por el sistema. Referencia a la tabla LOG_TABLE	No	Yes
IDEXCHANGE	int		identificador que referencia al intercambio al que hace referencia el registro.		
IDLOG_OPERATION_TYPE	smallint	NOT NULL	Código del tipo de acción o procesamiento que almacena el registro. F.K. que referencia a la tabla LOG_OPERATION_TYPE	No	Yes
DETAIL	varchar(100)	NULL	Detalles de la acción u operación registrada.	No	No

Table Name: LOG_OPERATION_TYPE

Table Comment: Tabla de referencia que contiene cada una de las acciones u operaciones individuales que se desean registrar en las tablas de log durante la ejecución del motor de búsqueda de intercambios compatibles.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDLOG_OPERATION_TYPE	smallint	NOT NULL	Primary Key de la tabla. Código numérico que identifica a cada tipo de acción o procesamiento que quiere ser registrado y clasificado en las tablas de log.	Yes	No
NAME	varchar(30)	NOT NULL	Nombre del tipo de acción o procesamiento que se quiere registrar en las tablas de log.	No	No

Table Name: LOG_TABLE

Table Comment: Tabla que almacena la información general de log para cada una de las ejecuciones del motor de búsqueda de intercambios compatibles.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDLOG	int	NOT NULL	P.K. identificador de los registros de la tabla. Código de log para cada una de las ejecuciones del sistema	Yes	No
DATE	datetime	NOT NULL	Fecha en la que se realizó la ejecución del procedimiento de calculo de intercambios por el motor ebarter.	No	No
RESUME	text	NULL	Texto con los datos del resumen de los procesos y acciones realizadas.	No	No

Table Name: NEIGHBORHOOD

Table Comment: Tabla de referencia que registra los diferentes vecindarios o distritos que son asociados a los usuarios del sistema

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDNEIGHBORHOOD	smallint	NOT NULL	P.K. identificador de cada uno de los vecindarios o barrios incluidos en el sistema e_barter.	Yes	No
NAME	varchar(30)	NOT NULL	Nombre del vecindario o barrio	No	No
IDCOUNTRY	tinyint	NULL	Pais en el que se encuentra el vecindario o barrio	No	Yes

Table Name: PERSONAL_WIZARD

Table Comment: Tabla de referencia que almacena la información de los diferentes asistentes disponibles para la personalización de la interfaz web por parte de los usuarios.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDPERSONAL_WIZARD	tinyint	NOT NULL	P.K. identificador del cada uno de los asistentes web que pueden ser seleccionados por el usuario.	Yes	No
NAME	varchar(30)	NOT NULL	Nombre del asistente	No	No
IMAGE_DATA	image	NULL	Datos de la imagen empleada por el asistente	No	No

Table Name: RESOURCE_CATEGORY

Table Comment: Tabla de referencia que contiene las diferentes categorías en las que podemos clasificar los recursos y productos gestionados en el sistema. Permite el anidamiento y referencias recursivas para poder establecer una jerarquía de clases y subclases entre las diferentes categorías.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDRESOURCE_CATEGORY	int	NOT NULL		Yes	Yes
NAME	varchar(20)	NOT NULL	Nombre de la categoría	No	No
MOTHER_CATEGORY	int	NULL	Referencia a otro registro de la misma tabla y que se corresponde con la categoría madre de la misma (ej. manzana y fruta). Proporciona un método recursivo para definir varios niveles de categorías. Si el valor de la columna es NULL significa que el registro no tiene categoría madre.	No	No
LEVEL	tinyint	NOT NULL	Numero que indica el nivel en la jerarquía establecida entre la categoría madre y cada uno de sus subcategorías. (valor comprendido entre 0 = categoría madre y N niveles de profundidad)	No	No



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 3: BASE DE DATOS

Página : 43/133

Table Name: RESOURCES

Table Comment: Tabla con todos los diferentes recursos dados de alta en el sistema e_barter. Cada uno de ellos se encuentra englobado dentro de una categoría para facilitar su clasificación y posterior selección desde las interfaces de aplicación. Solo es posible realizar peticiones de intercambios sobre los objetos contenidos en esta tabla. El mantenimiento de esta tabla se realiza solo por el usuario administrador.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
COST	numeric(9,2)	NULL	Campo que almacena el coste de referencia asignado por el sistema para el calculo de posibles gastos de transacción.	No	No
IDRESOURCE	int	NOT NULL	Primary Key de la tabla. Código numérico que identifica a cada uno de los recursos incluidos en el sistema.	Yes	No
IDRESOURCE_CATEGORY	int	NOT NULL	Código que identifica a la categoría a la que pertenece el recurso. F.K. que referencia a la tabla RESOURCE_CATEGORY	No	Yes
NAME	varchar(30)	NOT NULL	Nombre del recurso.	No	No

Table Name: SYSTEM_PARAMETER

Table Comment: Esta tabla contiene los diferentes parámetros de configuración y limitaciones del sistema. Permite regular de forma simple diferentes aspectos del funcionamiento del sistema.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
SLIDE_TIME	int	NULL	Tiempo asignado a cada Slide de procesamiento para el uso por agentes distribuidos en los que cada uno solo gestionan sus recursos.	No	No
IDSYSTEMPARAMETER	Int	NOT NULL	Pk. de la tabla.	Yes	No
MAX_USER_NUMBER	Int	NULL	Numero máximo de usuarios que admite el sistema. Permite establecer un limite atendiendo a los requerimientos de procesamiento y memoria requeridos por el algoritmo gestor de ebarter.	No	No
BOUNDARY	Int	NOT NULL	Numero máximo del nivel de colaboraciones para los evalúa el algoritmo de intercambio. Permite establecer un limite en la profundidad de evaluación y por tanto del tiempo de procesamiento.	No	No
MAX_DELAY_CONFIRM	int	NULL	Tiempo máximo de espera en la confirmación y respuesta a los intercambios propuestos por el sistema. Permite que un intercambio no confirmado no bloquee indefinidamente la información relacionada del sistema.	No	No
LOGS_PATH	varchar(255)	NOT NULL	Directorio en donde se almacenan los ficheros de LOG generados por el motor de intercambios	No	
JAVA_PATH	varchar(255)	NOT NULL	Directorio en donde se encuentra la aplicación Java que implementa el motor de intercambios.	No	No
URL_WEB	varchar(128)	NOT NULL	Dirección URL establecida para el cliente WEB.	No	No

Table Name: USER_LOGIN

Table Comment: Tabla que contiene los datos de identificación para el acceso vía cliente web para los diferentes usuarios registrados. Relacionada con la tabla EBARTER_USER en una relacion 1:1. Los datos se almacenan por separado por cuestiones de seguridad.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDUSER	int	NOT NULL	P.K. de la tabla. identificador único para cada usuario del sistema. Proviene de la relacion con la tabla EBARTER_USER	Yes	Yes
LOGIN	varchar(20)	NOT NULL	Login del usuario. Clave alternativa que también identifica a cada uno de los usuarios del sistema	No	No
PASSWORD	varchar(10)	NOT NULL	Password de conexión y autenticación del usuario	No	No
QUESTION	varchar(50)	NULL	Pregunta alternativa para solucionar problemas de olvido de password.	No	No
RESPONSE	varchar(50)	NOT NULL	Respuesta al campo QUESTIONB	No	No

Table Name: USER_STATUS

Table Comment: Tabla de referencia que contiene los diferentes posibles estados en los que se pueden encontrar los usuarios registrados en el sistema e_barter. Permite tener una calificación entre usuarios activos, dados de baja, etc.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDUSER_STATUS	tinyint	NOT NULL	P.K. identificador de los diferentes estados en los que se puede encontrar un usuario registrado en el sistema e_barter.	Yes	No
NAME	varchar(20)	NOT NULL	Nombre descriptivo del estado de un usuario registrado en el sistema e_barter	No	No

Table Name: V_RESOURCE_LIST

Table Comment: Vista que nos proporciona los 100 primeros recursos casados con sus categorías y en un formato que facilita la gestión y visualización de los mismos por parte del cliente web.



5 Desencadenadores y Procedimientos de Almacenado.

Como ya se menciona en la introducción sobre la base de datos, en ella se encuentran implementadas un conjunto de funcionalidades del sistema eBarter a través de una serie de procedimientos de almacenado y desencadenadores. (Código embebido en la base de datos implementado mediante Transact SQL).

Una utilidad añadida a la utilización de los procedimientos de almacenado es la centralización de cierto código que puede ser usado por los diferentes módulos del sistema (web, motor de intercambios y programa de administración). Esto permite independizar estas operaciones de las estructuras internas de las tablas y permite un mejor mantenimiento sobre posibles cambios futuros.

En este apartado se realiza la descripción básica de las funcionalidades implementadas a través de este código, así como un completo catalogo de dichos procedimientos de almacenado.

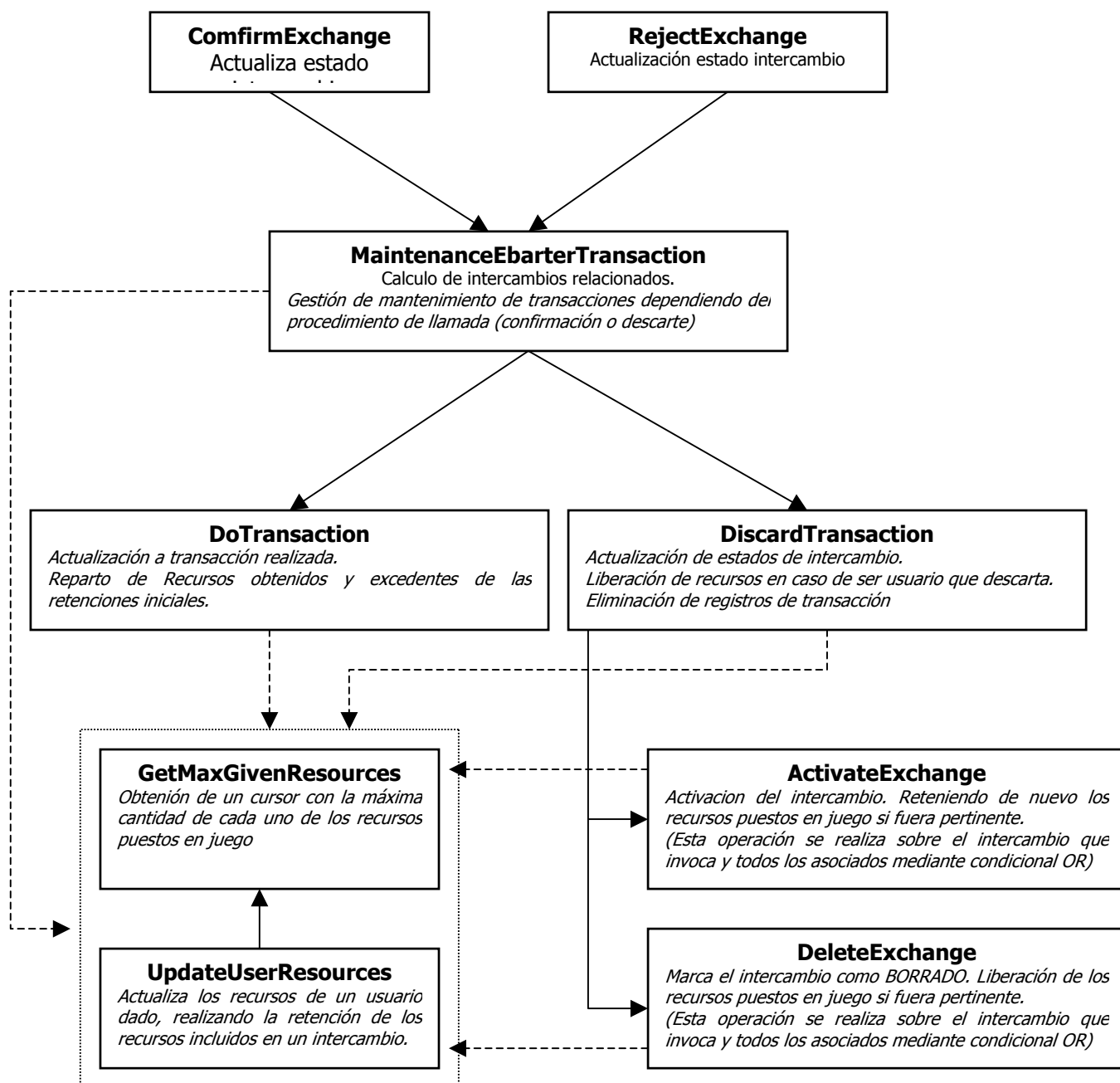
5.1 Descripción general y Principales funcionalidades.

Las principales funcionalidades implementadas son :

1. Obtención de datos. A través de este conjunto de procedimientos de almacenado obtenemos datos del sistema para ser usados por las aplicaciones.
 - a. Obtención de recursos de un usuario.
 - b. Obtención de intercambios, pudiendo realizar filtros por un estado determinado o por la pertenencia a un usuario dado.
2. Manipulación e inserción de datos. En este grupo tenemos
 - a. Inserción de usuarios
 - b. Actualización de datos de conexión y comprobaciones pertinentes.
 - c. Inserción de registros de log.
 - d. Inserción de recursos (del sistema, asignados a un usuario o asignados a un intercambio).
3. Gestión de los recursos de los usuarios. Entre las operaciones más relevantes se encuentran :
 - a. Calculo y retención de recursos puestos en juego en un intercambio (incluyendo todos los intercambios relacionados).
 - b. Devolución de recursos al usuario cuando un intercambio es descartado.
 - c. Adjudicación de los recursos obtenidos por un usuario a través de un intercambio realizado. (incluyendo recursos puestos en juego y que no han sido cedidos).
4. Control en la transición de estados de los intercambio incluidos en el sistema, incluyendo operaciones tan relevantes como son:
 - a. Confirmación de intercambios propuestos
 - b. Rechazo de intercambios propuestos
 - c. Re-Activación de intercambios. (Intercambios descartados con anterioridad).
 - d. Eliminación de Intercambios.
5. Tratamientos adicionales de soporte a aplicaciones. En este grupo se incluye la gestión transaccional y uso de tablas temporales que facilitan la inserción de intercambios. Las funciones implementadas más importantes son:
 - a. Creación de tablas temporales.
 - b. Confirmación de datos de intercambio y traspaso a tablas de producción real.
 - c. Destrucción de tablas temporales.



Cabe destacar que los procedimientos de almacenado se utilizan entre si facilitando la implementación de los mismos. A continuación se presenta un diagrama funcional sobre la interrelación de procedimientos de almacenado para las operaciones más relevantes y complejas, como son, la confirmación y el rechazo de los intercambios propuestos.





5.2 Catalogo Procedimientos de Almacenado.

A continuación se muestra un extracto de todos los procedimientos de almacenados incluidos en la base de datos del sistema eBarter.

```
Name      : sp_app_GetAppParameter          Version : 1.0.0          Date:01/05/03
Parameter : IN -> @IDAPP = Código de la aplicación.
           OUT -> @MULTILENGUAL valor booleano que indica si se debe aplicar soporte multi-idioma
                  @APPLYPROFILE valor booleano que indica si se debe aplicar soporte de perfiles
                  de usuario
Function  : Dada un código de aplicación devuelve los parámetros de si se debe aplicar el soporte
            de gestión de perfiles y multi-idioma.
Comment  : Para uso desde la aplicación de administración

/ ----- /

Name      : sp_app_GetAppUserConfigurationVersion : 1.0.0          Date:01/05/03
Parameter : IN -> @IDUSER = Código de usuario administrador.
           OUT -> @LANGUAGE valor entero que indica el idioma para el usuario
                  (si hay soporte multi-idioma)
                  @SHOWMAINMENU valor booleano que indica si se visualiza la barra de menu principal.
                  @SHOWTREETMENU valor booleano que indica si se visualiza el menu tree_view
                  @SHOWOBARMENU valor booleano que indica si se visualiza la barra tipo outlook
                  @AUTOHIDE valor booleano que indica la aplicación funciona en modo autohide.
Function  : Dada un código de usuario devuelve los parámetros de personalización de la aplicación
            para dicho usuario.
Comment  : Para uso desde la aplicación de administración

/ ----- /

Name      : sp_app_SetAppUserConfigurationVersion : 1.0.0          Date:01/05/03
Parameter : IN -> @IDUSER = Código de usuario administrador.
           @LANGUAGE valor entero que indica el idioma para el usuario
                  (si hay soporte multiidioma)
                  @SHOWMAINMENU valor booleano que indica si se visualiza la barra de menu principal.
                  @SHOWTREETMENU valor booleano que indica si se visualiza el menu tree_view
                  @SHOWOBARMENU valor booleano que indica si se visualiza la barra tipo outlook
                  @AUTOHIDE valor booleano que indica la aplicación funciona en modo autohide.
Function  : Dada un código de usuario devuelve los parámetros de personalización de la
            aplicación para dicho usuario.
Comment  : Para uso desde la aplicación de administración

/ ----- /

Name      : sp_eb_ActiveExchange             Version : 1.0.0          Date:01/05/03
Parameter : IN -> IDEXCHANGE intercambio a activar
           OUT -> RESULT código de resultado y control de la operación de borrado
Function  : Elimina un intercambio de la tabla EXCHANGE actualizando la mochila del usuario
            adecuadamente. Si el intercambio estaba relacionado mediante condicional OR se
            elimina de la relación, si solo queda un exchange en la relación se elimina la
            relación OR.
Comment  :

/ ----- /

Name      : sp_eb_AddeBarterTransaction       Version : 1.0.0          Date:01/05/03
Parameter : Intercambios Origen y Destino, Recurso dado y Cantidad del mismo.
Function  : Insercion en la tabla EBARTER_TRANSACTION del registro pertinente
            y actualización del intercambio origen al estado de PROPUESTA EBARTER
Comment  : Los intercambios se actualizan en cada inserción. Para optimizar el
            numero de actualizaciones Se puede emplear la inserción en una tabla
            temporal y una gestión global de los mismos.

/ ----- /

Name      : sp_eb_AddeBarterUser              Version : 1.0.0          Date:01/05/03
Parameter : Datos de identificación del usuario, excepto identificador interno
            y fecha alta que son generados automáticamente por el sistema
Function  : Insertar un nuevo registro de usuario en las tablas EBARTER_USER y USER_LOGIN
Comment  :

/ ----- /

Name      : sp_eb_AddLogDetail                Version : 1.0.0          Date:01/05/03
Parameter : IN -> IDLOG identificador numérico del LOF
            IDEXCHANGE identificador del intercambio al que hace refer.
            IDPOERATION identificador del tipo de operación a registrar
            DETAIL Detalles y texto adicional a insertar
```



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Proyecto S.I. / e-Barter. (Año 2002/03)

Sección 3: BASE DE DATOS

Página : 47/133

Function : Insertar un nuevo registro en la tabla de detalles de log
LOG_EXCHANGE_DETAIL con los datos pasados como parámetros
Comment :

/ ----- /
Name : sp_eb_AddResourceToExchange Version : 1.0.0 Date:01/05/03
Parameter : IN --> @IDEXCHANGE identificación del intercambio
@IDRESOURCE identificador del recurso
@AMOUNT cantidad del recurso a añadir o insertar
@TYPE : Campo que indica si es un recurso a dar u obtener (0 Dar / 1 Obtener)
@TEMP : indica el uso de tablas temporales en la inserción de intercambios.
OUT--> @RESULT código de retorno sobre el resultado de la operación
(0= OK)
Function : Añade recursos a un intercambio dado. Si el recurso ya esta incluido
añade la cantidad indicada, sino, inserta un nuevo registro con los
datos incluidos en los parámetros de entrada.
Comment :

/ ----- /
Name : sp_eb_AddUserResource Version : 1.0.0 Date:01/05/03
Parameter : IN --> IDUSER identificación del usuario
IDRESOURCE identificador del recurso
AMOUNT cantidad del recurso a añadir o insertar
OUT --> RESULT código de retorno sobre el resultado de la operación (0= OK)
Function : Añade recursos a la mochila del usuario. Si el recurso ya esta
incluido en la mochila añade la cantidad indicada, sino, inserta
un nuevo registro con los datos incluidos en los parámetros de entrada.
Comment :

/ ----- /
Name : sp_eb_CalculatedCategoryLevel Version : 1.0.0 Date:01/05/03
Parameter : IN -> IDEXCHANGE identificador de la categoría sobre la que se quiere obtener el nivel
OUT -> Nivel para esa categoría.
Function : Función que me determina el nivel de profundidad para una categoría
dada (jerarquía de niveles entre una categoría y N categorías madre)
Comment : Utilizada desde Trigger en la tabla RESOURCE_CATEGORY.

/ ----- /
Name : sp_eb_CalculatedTransactionCost Version : 1.0.0 Date:01/05/03
Parameter : IN -> IDEXCHANGE identificador del intercambio.
Function : Calcula el coste de una transacción en función de los recursos dado
y recibidos. Actualiza dicho valor en el campo COST de la tabla EXCHANGE
Comment :

/ ----- /
Name : sp_eb_ConfirmExchange Version : 1.0.0 Date:01/05/03
Parameter : IN -> IDEXCHANGE
OUT -> RESULT código numérico de resultado de operación (0=OK)
Function : Marca el intercambio que se pasa como parámetro como confirmado
por el usuario e invoca al procedimiento de mantenimiento
general de la tabla de transacciones propuestas.
Comment :

/ ----- /
Name : sp_eb_ConfirmTemporalExchangeTables Version : 1.0.0 Date:01/05/03
Parameter : OUT--> RESULT código de retorno sobre el resultado de la operación(0= OK)
Function : Procedimiento que actualiza adecuadamente los identificadores de
las tablas temporales utilizadas en la inserción de intercambios
y trasfiere la información a las tablas reales del sistema.
Comment : La inserción de intercambio a través de las tablas temporales
permite mejorar la concurrencia y facilita el control de transacciones desde la app.

/ ----- /
Name : sp_eb_CreateLog Version : 1.0.0 Date:01/05/03
Parameter :
Function : Inserta un nuevo registro de log con los datos de resumen
pasados en el parámetro de entrada y devuelve el identificador asignado al mismo
Comment :

/ ----- /
Name : sp_eb_CreateTemporalExchangeTables Version : 1.0.0 Date:01/05/03
Parameter : OUT--> RESULT código de retorno sobre el resultado de la operación (0= OK)
Function : Crea el conjunto de tablas temporales necesarias para la Inserción
OFFLINE de intercambios
Comment : La inserción de intercambio a través de las tablas temporales permite
mejorar la concurrencia y facilita el control de transacciones desde la app.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 3: BASE DE DATOS

Página : 48/133

```
----- /
Name      : sp_eb_DeleteEBarterUser          Version : 1.0.0          Date:01/05/03
Parameter : IN -> IDUSER: identificador del usuario y MODE
              (0:Automático ; 1: Borra, 2: Dar de baja)
              OUT -> RESULT código de retorno de la operación (0=ok)
Function   : Borra el registro de usuario o da de baja el usuario con identificador
              pasado como parámetro dependiendo del modo de llamada (parámetro MODE).
Comment   :
----- /
```

```
----- /
Name      : sp_eb_DeleteEbarterTransactionVersion : 1.1.0          Date:01/06/03
Parameter : IN -> IDEXCHANGE intercambio de referencia para borrar los datos de
              transacciones propuestas. (registros de la tabla EBARTER_TRANSACTION)
              relac. con el intercambio
              OUT -> RESULT código numérico de resultado de operación (0=OK)
Function   : Elimina la información contenida en la tabla EBARTER_TRANSACTION
              relacionada con el intercambio que se pasa como parámetro.
Comment   :
----- /
```

```
----- /
Name      : sp_eb_DeleteExchange             Version : 1.0.0          Date:01/05/03
Parameter : IN -> IDEXCHANGE intercambio a borrar
              OUT -> RESULT código de resultado y control de la operación de borrado
Function   : Elimina un intercambio de la tabla EXCHANGE actualizando la mochila
              del usuario adecuadamente.
Comment   :
----- /
```

```
----- /
Name      : sp_eb_CreateLog                   Version : 1.0.0          Date:01/05/03
Parameter : IN --> LOG identificador del LOG a borrar
              OUT --> Resultado de la operación (0=ok)
Function   : Procedimiento que borra las entradas del LOG con identificador
              que se pasa como parámetro
Comment   :
----- /
```

```
----- /
Name      : sp_eb_DeleteResource              Version : 1.0.0          Date:15/06/03
Parameter : IN -> IDRESOURCE identificador del intercambio a borrar
              OUT -> RESULT código de resultado y control de la operación de borrado(0=OK)
Function   : Elimina un recurso de la tabla RESOURCES. Si el recurso esta siendo
              utilizado por algún usuario o intercambio retorna código de error 1,
              indicando que la operación no puede realizarse. (Referencias de integridad)
Comment   :
----- /
```

```
----- /
Name      : sp_eb_DeleteResourceCategory      Version : 1.0.0          Date:15/06/03
Parameter : IN -> IDCATEGORY identificador del intercambio a borrar
              OUT -> RESULT código de resultado y control de la operación de borrado(0=ok)
Function   : Elimina un recurso de la tabla RESOURCE_CATEGORY. Si la categoría tiene algún
              recurso relacionado se retorna código de error 1, indicando que la operación no
              puede realizarse. (Referencias de integridad)
Comment   :
----- /
```

```
----- /
Name      : sp_eb_DeleteUserResource          Version : 1.0.0          Date:01/05/03
Parameter : IN -> IDUSER identificación del usuario
              IDRESOURCE identificador del recurso
              OUT -> RESULT código de retorno sobre el resultado de la operación (0= OK)
Function   : Elimina el recurso que se pasa como parámetro de la mochila del usuario.
Comment   :
----- /
```

```
----- /
Name      : sp_eb_DiscardTransaction          Version : 2.0.0          Date:01/05/03
Parameter : IN -> IDEXCHANGE identificador del intercambio a descartar (junto con sus relacionados)
              DISCARD_MODE: Indica la operación a realizar en el mantenimiento
              de descartes (0= Reactiva Intercambios / 1 Elimina Intercambios)
Function   : Actualiza la base de datos atendiendo a que la propuesta dada para el intercambio
              que se pasa como parámetro es desechada.
Comment   : Esta acción influye en el estado del intercambio y la actualización
              de los recursos del usuario que fueron reservados para el mismo.
----- /
```

```
----- /
Name      : sp_eb_DoTransacion                Version : 1.0.0          Date:01/05/03
Parameter :
Function   : Realiza las actualizaciones necesarias tras la confirmación de que el intercambio
              pasado como parámetro puede ser realizado (todas las partes implicadas hanconfirmado).
----- /
```



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Proyecto S.I. / e-Barter. (Año 2002/03)

Sección 3: BASE DE DATOS

Página : 49/133

Comment : Actualiza tanto el estado del intercambio como la mochila del usuario (recursos reales dado y recursos recibidos).

Name : sp_eb_DropTemporalExchangeTables Version : 1.0.0 Date:01/05/03

Parameter : OUT--> RESULT código de retorno sobre el resultado de la operación (0= OK)

Function : Elimina el conjunto de tablas temporales creadas para la inserción OFFLINE de intercambios

Comment : La inserción de intercambio a través de las tablas temporales permite mejorar la concurrencia y facilita el control de transacciones desde la aplicación.

Name : sp_eb_GetExchangesByStatus Version : 1.0.0 Date:01/05/03

Parameter : IN -> IDSTATUS filtro de un estado determinado para los intercam.
OUT -> Cursor con la lista de intercambios

Function : procedimiento de almacenado que devuelve un cursor con todos los intercambios contenidos en la tabla EXCHANGE que se encuentran en el estado indicado en el parámetro de entrada @IdStatus

Comment : Si no se pasa ningún valor en el parámetro IDSTATUS retorna todos los intercambios incluidos en la tabla.

Name : sp_eb_GetGivenResources Version : 1.0.0 Date:01/05/03

Parameter : IN -> intercambio del que se quieren obtener los recursos dados
OUT -> Cursor que contiene la lista de recursos y cantidades dadas.

Function : Devuelve un cursor con los recursos dados en el intercambio que se pasa como parámetro

Comment :

Name : sp_eb_GetIDResourceNumber Version : 1.0.0 Date:01/05/03

Parameter : IN -> @TABLENAME Nombre de la tabla del que se quiere obtener el identif.
(RESOURCES O RESOURCE_CATEGORY)

@MOTHER_CATEGORY identificador de la categoría madre para asignar numeración.

OUT -> identificador valido (MAX+1)

Function : Devuelve un identificador utilizable para la tabla que se pasa como parámetro atendiendo a los rango previos de las categorías madre.

Comment :

Name : sp_eb_GetIDnumber Version : 1.0.0 Date:01/05/03

Parameter : IN -> Nombre de la tabla del que se quiere obtener el identif.
OUT -> identificador valido (MAX+1)

Function : Devuelve un identificador utilizable para la tabla que se pasa como parámetro

Comment :

Name : sp_eb_GetMaxResourceToGive Version : 1.0.0 Date:01/05/03

Parameter : IN -> IDEXCHANGE identificador del intercambio de referencia.

OUT -> RESOURCE_CURSOR Cursor con los recursos y cantidad máxima de ellos puestos en juego en un intercambio y todos sus relacionados.

Function : Retorna un cursos que contiene todos los recursos con la máxima cantidad de ellos que se encuentran incluidos en un intercambio y todos los relacionados a través de una condicional OR

Comment :

Name : sp_eb_GetObtainedResources Version : 1.0.0 Date:01/05/03

Parameter : IN -> intercambio del que se quieren conocer los recursos obtenidos

OUT -> Cursor que contiene la lista de recursos y cantidades obtenidas .

Function : Devuelve un cursor con los recursos obtenidos en el intercambio que se pasa como parámetro

Comment :

Name : sp_eb_GetExchangesByStatus Version : 1.0.0 Date:01/05/03

Parameter : IN -> IDSTATUS filtro de un estado determinado para los intercambios.

OUT -> Cursor con la lista de intercambios

Function : procedimiento de almacenado que devuelve un cursor con todos los intercambios contenidos en la tabla EXCHANGE que pertenecen al usuario indicado en el parámetro de entrada @IDUSER.

Usamos el parámetro @ExchangeStatus para realizar un filtro sobre solo los intercambios que se encuentran en un determinado estado, si el valor es NULL no se realiza filtro por esta propiedad

Comment :

Name : sp_eb_GetUserResource Version : 1.0.0

Date:01/05/03



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Proyecto S.I. / e-Barter. (Año 2002/03)

Sección 3: BASE DE DATOS

Página : 50/133

```
Parameter : IN -> IDUSER identificador de usuario
              MODE (0 solo coge datos de la mochila, <>0 se refiere a un código de intercambio
              sobre el que se quieren sumar los recursos incluidos en el mismo)
              OUT -> @RESULT código de retorno operación (0=ok)
              @RESOURCE_CURSOR cursor de recursos
Function : Retorna un cursos con los recursos que tiene el usuario que se pasa como parámetro.
Comment :
```

/*-----

```
Name       : sp_eb_GetUserResourceAmount      Version : 1.0.0      Date:01/05/03
Parameter : IN -> IDUSER : identificador del usuario
              IDRESOURCE : identificador del recurso.
              OUT -> Cantidad del recurso para el usuario pasados como parámetro entrada
Function : Devuelve la cantidad del recurso que tiene un usuario dado
Comment : Devuelve 0 si no hay registro para los datos dados.
```

/-----

```
Name       : sp_eb_GetUserResourceWEB      Version : 1.0.0      Date:01/05/03
Parameter : IN -> IDUSER identificador de usuario
              MODE (0 solo coge datos de la mochila, <>0 se refiere a un código de intercambio
              sobre el que quieren sumar los recursos incluidos en el mismo
              OUT -> @RESULT código de retorno operación (0=ok)
Function : Retorna un cursos con los recursos que tiene el usuario que se pasa como parámetro.
Comment :
```

/-----

```
Name       : sp_eb_MaintenanceEBarterTransacion Version : 1.2.0 Date:01/05/03
Parameter : IN-> IDEXCHANGE : Intercambio que invoca al procedimiento porque ha sufrido una
              modificación de estado. (Confirmacion o rechazo de propuesta).
              TYPE : Indica que tipo de evento a generado la invocación del
              procedimiento. (0=confirmacion / 1=rechazo)
              DISCARD_MODE: Indica la operación a realizar en el mantenimiento de descartes.
              (0= Reactiva Intercambios / 1 Elimina Intercambios)
              OUT -> RESULT indica resultado de operación del sp. (0=OK)
Function : Procedimiento general de mantenimiento de la tabla de transacciones
              propuestas por el sistema. Se encarga de comprobar el estado de todos los
              intercambios relacionados con el que se pasa como parámetro. En caso de que
              todas las transacciones han sido confirmadas y realiza las operaciones de
              actualización oportunas en los intercambios y recursos de usuarios implicados.
Comment : Usa procedimientos DoTransaction y DiscardTransaction : que se encargan
              finalmente de realizar las actualizaciones oportunas en cada intercambio
              implicado y mochila de recursos de sus usuarios.
```

/-----

```
Name       : sp_eb_RejectExchange      Version : 1.2.0      Date:01/05/03
Parameter : IN -> IDEXCHANGE identificador del intercambio de referencia
              DISCARD_MODE: Indica la operación a realizar en el mantenimiento de descartes
              (0= Reactiva Intercambios / 1 Elimina Intercambios)
              OUT -> RESULT código numérico de resultado de operación (0=OK)
Function : Marca el intercambio que se pasa como parámetro como RECHAZADO por el usuario e invoca
              al procedimiento de mantenimiento general de la tabla de transacciones propuestas.
Comment :
```

/-----

```
Name       : sp_eb_UpdateUserLogin      Version : 1.0.0      Date:01/05/03
Parameter : Datos de identificación del usuario (IDUSER)
              + Datos de Acceso (LOGIN, PASSWORD, QUESTION, RESPONSE)
Function : Procedimiento que actualiza los datos de acceso de un usuario dato en la tabla USER_LOGIN
Comment :
```

/-----

```
Name       : sp_eb_UpdateUserResources      Version : 1.0.0      Date:01/05/03
Parameter : IN -> IDEXCHANGE intercambio de referencia que invoca la ejecución del procedimiento.
Function : procedimiento de almacenado que actualiza los recursos incluidos en la mochila del
              usuario al que pertenece el intercambio que se pasa como parámetro.
Comment : Utiliza el procedimiento sp_eb_GetMaxResourceToGive para obtener los recursos
              implicados en el intercambio.
```

/-----

```
Name       : sp_eb_UpdateUserResourcesWEB      Version : 1.0.0      Date:01/05/03
Parameter : IN -> IDEXCHANGE intercambio de referencia que invoca la ejecución del
              procedimiento.
              IDCONDITIONAL identificador de relación condicional
              OUT -> Código de operación realizada (0=ok)
Function : procedimiento de almacenado que actualiza los recursos incluidos en la mochila del
              usuario al que pertenece el intercambio que se pasa como parámetro.
Comment : Utiliza el procedimiento sp_eb_GetMaxResourceToGive para obtener los recursos
              implicados en el intercambio. (modif. particular de sp_eb_UpdateUserResources
              para la utilización desde la web)
```




5.3 Desencadenadores.

Para finalizar el apartado referente al código incluido dentro del propio sistema gestor de la base de datos cabe destacar la existencia de 2 triggers, que facilitan el calculo automático del nivel para una categoría de recurso y el calculo del coste a imputar para una transacción proporciona por el sistema (en función de los recursos obtenidos mediante ella).

A continuación se presentan dichos triggers.

```
/* *****  
ACTUALIZACION DEL IMPORTE DE TRANSACCION EN TABLA EXCHANGE EN FUNCION DE LOS  
RECURSOS OBTENIDOS  
ACTIVACION --> ACTUALIZACION DEL INTERCAMBIO A ESTADO 2  
***** */  
CREATE TRIGGER tr_SetTransactionCost ON [dbo].[EXCHANGE]  
AFTER UPDATE  
AS  
BEGIN  
    DECLARE @RESULT INT  
    DECLARE @IDEXCHANGE INT  
    DECLARE @STATUS INT  
  
    IF UPDATE (IDEXCHANGE_STATUS)  
    BEGIN  
        SELECT @STATUS= IDEXCHANGE_STATUS, @IDEXCHANGE= IDEXCHANGE FROM INSERTED  
        IF @STATUS=2  
        BEGIN  
            EXECUTE sp_eb_CalculatedTransactionCost @IDEXCHANGE, @RESULT OUTPUT  
        END  
    END  
END  
GO  
  
/* *****  
CALCULO Y ACTUALIZACION DEL NIVEL CORRESPONDIENTE A UNA NUEVA CATEGORIA EN  
FUNCION DEL VALOR DE LA CATEGORIA MADRE.  
ACTIVACION --> INSERCIÓN DE UNA NUEVA CATEGORIA DE RECURSOS.  
***** */  
CREATE TRIGGER INSERT_CATEGORYLEVEL ON dbo.RESOURCE_CATEGORY  
AFTER INSERT  
AS  
-- declaracion de variables locales  
DECLARE @LEVEL INT  
DECLARE @RESULT INT  
DECLARE @IDRESOURCE_CATEGORY INT;  
-- seleccionamos categoria madre del nuevo registro  
SELECT @IDRESOURCE_CATEGORY = IDRESOURCE_CATEGORY FROM inserted  
-- llamada a procedimiento de almacenado que calcula la profundidad  
EXECUTE sp_eb_CalculatedCategoryLevel @IDRESOURCE_CATEGORY, @LEVEL = @RESULT OUTPUT  
-- actualizamos el valor del nivel para el nuevo registro  
UPDATE RESOURCE_CATEGORY SET LEVEL = @RESULT  
WHERE IDRESOURCE_CATEGORY = @IDRESOURCE_CATEGORY  
GO
```



6 Ampliación soporte Multi-Idioma (Aplicación Administrador)

6.1 Descripción de la ampliación.

Adicionalmente se ha incluido una serie de tablas adicionales que si bien no tienen funcionalidad en cuanto al servicio de intercambio de recursos, nos permiten ajustar la aplicación de administración para que funcione sobre diferentes idiomas.

Esto ha sido realizado debido al interés del tutor del proyecto de que el idioma empleado sea el ingles. Permitiendo correcciones futuras en los mensajes y traducciones así como la adaptación a otros lenguajes de alta utilización como es el Francés.

El modelo incluye campos de registro para el ESPAÑOL, INGLES, FRANCES y otro lenguaje de libre selección.

Una vez que los diferentes elementos están registrados en las tablas que se presentan a continuación, la aplicación de administración realiza una lectura de las mismas antes de mostrar las diferentes pantallas adaptando las propiedades de visualización (CAPTION, TEXT y HINT) al idioma seleccionado.

El sistema presentado gestiona el idioma de los diferentes mensajes dados al usuario a través de la tabla S_MESSAGES, ya que no se ha completado la introducción de datos en el resto de las tablas que registran todos los objetos. (si bien se proporciona utilidad para realizar esta carga y posterior traducción).

Observación.

Para activar el funcionamiento multi-idioma a nivel completo se debe marcar el campo MULTILENGUAL de la tabla S_APLICATION a true.



6.2 Diagrama del modelo.

La ampliación de tablas en el modelo para el soporte multi-idioma se presentan en el siguiente diagrama.

Project...: E-BARTER

Versión...: 1.2.1.

: SQL-SERVER

Date.....: 01/03/02

APLIACION ADMINISTRACION SOPORTE MULTI-IDIOMA.

U_USER

IDUSER: smallint NOT NULL
NAME: varchar(20) NULL
SURNAME: varchar(30) NULL
LOGIN: varchar(25) NULL
PASSWORD: varchar(10) NULL
ACTIVE: bit NOT NULL
BEGINDATE: smalldatetime NULL
ENDDATE: smalldatetime NULL
TITLE: varchar(30) NULL
EMAIL: varchar(64) NULL
IDLANGUAGE: tinyint NOT NULL
SHOWMAINMENU: bit NOT NULL
SHOWTREEVIEW: bit NOT NULL
SHOWOUTLOOKBAR: bit NOT NULL
APPLICATION_AUTOHIDE: bit NOT NULL

S_MESSAGE

IDMESSAGE: int NOT NULL
SPANISH: varchar(1984) NULL
FRENCH: varchar(1984) NULL
ENGLISH: varchar(1984) NULL
LANGUAGE: varchar(1984) NULL
MESSAGETYPE: tinyint NOT NULL

S_APPLICATION

IDAPPLICATION: tinyint NOT NULL
NAMEAPPLICATION: varchar(255) NOT NULL
DESCAPPLICATION: varchar(1984) NULL
MULTILENGUAL: bit NOT NULL

S_FORM

IDFORM: int IDENTITY(1,1)
IDAPPLICATION: tinyint NOT NULL (FK)
NAMEFORM: varchar(255) NOT NULL
DESCFORM: varchar(255) NULL

S_FORMOBJECT

IDFORMOBJECT: int IDENTITY(1,1)
IDFORM: int NOT NULL (FK)
NAMEFORMOBJECT: varchar(255) NOT NULL
DESCFORMOBJECT: varchar(255) NULL
WIDTH: int NULL
LENGTH: int NULL
IDMASK: int NULL (FK)
IDMASKPROPERTY: int NULL (FK)
IDTRANSLATIONPROPERTY: int NULL (FK)
IDTRANSLATIONSTRING: int NULL (FK)
IDTRANSLATIONSTRINGHINT: int NULL (FK)
OBJECTTYPE: tinyint NOT NULL

S_MASK

IDMASK: int IDENTITY(1,1)
MASKNAME: varchar(255) NULL
MASK: varchar(255) NULL

S_TRANSLATIONSTRING

IDTRANSLATIONSTRING: int IDENTITY(1,1)
SPANISH: varchar(255) NULL
FRENCH: varchar(255) NULL
ENGLISH: varchar(255) NULL
LANGUAGE: varchar(255) NULL

S_PROPERTY

IDPROPERTY: int IDENTITY(1,1)
PROPERTYTYPE: int NULL
PROPERTYNAME: varchar(255) NULL



6.3 Catalogo-Referencia de tablas.

Table Name	Table Comment
S_APPLICATION	Tabla de registro de la aplicación de administración. Permite proporcionar el mismo soporte multiidioma a diferentes aplicaciones de forma independiente.
S_FORM	Tabla que almacena todas las ventanas de la aplicación.
S_FORMOBJECT	Tabla que almacena los objetos pertenecientes a un formulario determinado. no es necesario reflejar todos los objetos, solo los que precisen algún procesamiento concreto (restricciones de uso, traducción de caption, mascarar...).)
S_MASK	Tabla que almacena las mascarar asociadas a una propiedad del objeto. Habrá una mascara diferente para cada clase de campo (teléfono, importe,...) que necesite ser formateado.
S_MESSAGE	Tabla donde se registran los mensajes del sistema. Permite el soporte multiidioma. Aunque se categoricen los tipos de mensaje por rangos, se le pone un identificador de tipo de mensaje (a revisar...)
S_PROPERTY	Tabla que almacena los nombres de propiedad sobre la que se tiene que aplicar las cadenas a visualizar y las mascarar. Se incluye debido a que la propiedad puede variar de un objeto a otro.
S_TRANSLATIONSTRING	
U_USER	Tabla general de registros de usuarios administradores del sistema Ebarter, utilizada para login y configuración de la aplicación de administración

6.4 Descripción física de tablas.

Table Name: S_APPLICATION

Table Comment: Tabla de registro de la aplicación de administración. Permite proporcionar el mismo soporte multiidioma a diferentes aplicaciones de forma independiente.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
MULTILENGUAL	bit	NOT NULL	Campo booleano que indica si la aplicación esta preparada para el soporte multiidioma para todos los objetos y componentes incluidos en la misma.	No	No
IDAPPLICATION	tinyint	NOT NULL	Clave de la tabla. Autoincremental	Yes	No
NAMEAPPLICATION	varchar(255)	NOT NULL	Nombre de la aplicación	No	No
DESCAPPLICATION	varchar(1984)	NULL	descripción de la aplicación	No	No

Table Name: S_FORM

Table Comment: Tabla que almacena todas las ventanas de la aplicación.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDFORM	int	IDENTITY	Clave de la tabla. Autoincremental.	Yes	No
NAMEFORM	varchar(255)	NOT NULL	Nombre del objeto formulario.	No	No
IDAPPLICATION	tinyint	NOT NULL	Aplicación a la que pertenece el formulario	No	Yes
DESCFORM	varchar(255)	NULL	Descripción del formulario	No	No

Table Name: S_FORMOBJECT

Table Comment: Tabla que almacena los objetos pertenecientes a un formulario determinado. no es necesario reflejar todos los objetos, solo los que precisen algun procesamiento concreto (restricciones de uso, traduccion de caption, mascarar...).

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDFORMOBJECT	int	IDENTITY	clave de la tabla. Autoincremental.	Yes	No
IDFORM		NOT NULL	Formulario al que pertenece el objeto	No	Yes
NAMEFORMOBJECT	varchar(255)	NOT NULL	Nombre del objeto	No	No
DESCFORMOBJECT		NULL	Descripción del objeto	No	No
WIDTH	int	NULL	longitud del objeto	No	No
LENGTH	Int	NULL	Longitud de edición del objeto	No	No
IDMASK	Int	NULL	identificador del campo que contiene la mascara a aplicar al objeto	No	Yes
IDMASKPROPERTY	Int	NULL	identificador del campo que contiene la propiedad (mask, editmask, displaymask...) que contiene la mascara a aplicar al objeto	No	Yes
IDTRANSLATIONPROPERTY	Int	NULL	identificador del campo que contiene la propiedad (caption, text, ...) que contiene la cadena a visualizar en el objeto	No	Yes
IDTRANSLATIONSTRING	Int	NULL	identificador de la cadena a visualizar en la propiedad del objeto (caption, title,text...) donde se introducirá la cadena.	No	Yes
IDTRANSLATIONSTRINGHINT	Int	NULL	identificador de la cadena a visualizar en el hint del objeto.	No	Yes
OBJECTTYPE	tinyint	NOT NULL	Tipo de objeto	No	No

Table Name: S_MASK



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de la base de datos y código Transac-SQL.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Proyecto S.I. / e-Barter. (Año 2002/03)

Sección 3: BASE DE DATOS

Página : 55/133

Table Comment: Tabla que almacena las mascara asociadas a una propiedad del objeto. Habra una mascara diferente para cada clase de campo (teléfono, importe,...) que necesite ser formateado.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDMASK	int	IDENTITY	clave de la tabla. Autoincremental.	Yes	No
MASKNAME	varchar(255)	NULL	Nombre del tipo de mascara	No	No
MASK	varchar(255)	NULL	definición de la mascara. se asigna al valor de la propiedad del componente.	No	No

Table Name: S_MESSAGE

Table Comment: Tabla donde se registran los mensajes del sistema. Permite el soporte multiidioma. Aunque se categoricen los tipos de mensaje por rangos, se le pone un identificador de tipo de mensaje (a revisar...)

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDMESSAGE	int	NOT NULL	Clave de la tabla. Autoincremental	Yes	No
SPANISH	varchar(1984)	NULL	Mensaje en español	No	No
FRENCH	varchar(1984)	NULL	Mensaje en frances	No	No
ENGLISH	varchar(1984)	NULL	Mensaje en ingles	No	No
LANGUAGE	varchar(1984)	NULL	Mensaje en idioma local	No	No
MESSAGETYPE	Tinyint	NOT NULL	tipo de mensaje (a revisar, si se tipifican por subrangos...)	No	No

Table Name: S_PROPERTY

Table Comment: Tabla que almacena los nombres de propiedad sobre la que se tiene que aplicar las cadenas a visualizar y las mascaras. Se incluye debido a que la propiedad puede variar de un objeto a otro.

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDPROPERTY	int	IDENTITY	clave de la tabla. Autoincremental.	Yes	No
PROPERTYTYPE	int	NULL	Tipo de propiedad.	No	No
PROPERTYNAME	varchar(255)	NULL	Nombre de la propiedad del objeto.	No	No

Table Name: S_TRANSLATIONSTRING

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
IDTRANSLATIONSTRING	int	IDENTITY	Clave de la tabla. Autoincremental.	Yes	No
SPANISH	varchar(255)	NULL	Texto a mostrar. En Español.	No	No
FRENCH	varchar(255)	NULL	Texto a mostrar. En frances.	No	No
ENGLISH	varchar(255)	NULL	Texto a mostrar. En Ingles.	No	No
LANGUAGE	varchar(255)	NULL	Texto a mostrar. En idioma adicional	No	No

Table Name: U_USER

Table Comment: Tabla general de registros de usuarios administradores del sistema Ebarter, utilizada para login y configuración de la aplicación de administración

Column Name	Datatype	Null Option	Description & Comments	Is PK	Is FK
ACTIVE	Bit	NOT NULL	Campo booleano que indica si el usuario esta activo	No	No
SHOWMAINMENU	Bit	NOT NULL	Mostrar barra de menu en la aplicación	No	No
SHOWTREEVIEW	Bit	NOT NULL	Mostrar TreeView Menu en la aplicación	No	No
SHOWOUTLOOKBAR	Bit	NOT NULL	Mostrar OutLook Bar menu en la aplicación.	No	No
APPLICATION_AUTOHIDE	Bit	NOT NULL	Campo booleano para activar el AutoHide del programa administrador al cambiar a otra aplicación.	No	No
IDUSER	smallint	NOT NULL	P.K. identificador interno del usuario	Yes	No
LOGIN	varchar(25)	NULL	Login del usuario usado para el acceso a la aplicación	No	
NAME	varchar(20)	NULL	Nombre del Empleado.	No	No
SURNAME	varchar(30)	NULL	Apellidos.	No	No
PASSWORD	varchar(10)	NULL	Password de acceso.	No	No
EMAIL	varchar(64)	NULL	e-mail del usuario administrador	No	No
IDLANGUAGE	tinyint	NOT NULL	Lenguaje de personalización de la aplicación.	No	No
TITLE	varchar(30)	NULL	Título o descripción adicional para el usuario administrador.	No	No
BEGINDATE	smalldatetime	NULL	Fecha de alta	No	No
ENDDATE	smalldatetime	NULL	Fecha de baja.	No	No



7 Instalación de la Base de Datos.

Una vez instalado el software correspondiente a SQL-Server 2000 (No incluido en el material suministrado) se deben ejecutar los script proporcionado con los cuales se crean todas las estructuras e inicializaciones de datos que se requieren para el correcto funcionamiento del sistema eBarter.

Estos SCRIPT pueden ser ejecutados desde la herramienta de consultas y sentencias SQL incluida en SQL-Server 2000

1. BASE DE DATOS.

Monta la base de datos para su utilización. Esta operación también puede realizarse libremente desde el administrador corporativo incluido en el propio SQL-SERVER.

Observación: Cambiar el path en donde se desea crear la base de datos.

2. DEFINICIÓN DE TABLAS, INDICES, FOREIGN KEYS y TRIGGERS

Creación de todas las estructuras para el almacenamiento y relacion de los datos dentro del sistema.

3. PROCEDIMIENTOS DE ALMACENADO.

Procedimientos de almacenados detallados con anterioridad en el documento.

4. DATOS INICIALES.

Carga un conjunto de datos iniciales sobre las tablas de referencia empleadas. (Posibles estados de los usuarios, intercambios, recursos iniciales, etc.)

5. FICHEROS DE PERMISOS.

Incluye una definición de los diferentes permisos de acceso para las estructuras de la base de datos EBARTER.

Observación: Los pasos 2 a 5 deben ser realizados sobre la base de datos EBARTER creada en el punto 1.



8 Apendices. Instalación y notas adicionales.

La información específica del código referente a la creación de los diferentes objetos incluidos en la base de datos no han sido incluidas en este documento, si bien, están disponibles en el directorio[.\DATABASE\SCRIPT].

A continuación se presenta una descripción y enlace a estos documentos.

8.1 Database Script. (SQL).

Se dispone del siguiente conjunto de Script en SQL y/o Transact SQL con la finalidad de proporcionar el código fuente de creación de todos los objetos incluidos en la base de datos

1. Schema (Base de datos y Usuario Administrado "ebadmin")

Contiene el script necesario para la creación de la base de datos y el usuario administrador de la misma (ebadmin), también utilizado desde la aplicación de administración que se presenta en la sección 4 de este mismo documento.

[\[Database Script\]](#)

2. Tablas, Relaciones de Integridad, Índices y Triggers

Contiene el conjunto de sentencias SQL para montar toda la estructura de tablas y relaciones necesarias para almacenar los datos requeridos por el sistema eBarter.

[\[Tablas Script\]](#)

3. Procedimientos de Almacenado.

Contiene el conjunto de sentencias SQL y Transact SQL correspondientes al conjunto de procedimientos de almacenado descritos con anterioridad en esta misma sección.

[\[StoreProcedure Script\]](#)

4. Datos de inicialización del sistema

Contiene el conjunto de datos que deben insertarse en las tablas del sistema eBarter para la inicialización del mismo y su correcto funcionamiento (ejemplo: definición de los estados de los usuarios y los intercambios).

[\[DefaultData Script\]](#)

[\[Permisos Script\]](#)

8.2 Herramienta de desarrollo

Para el análisis y diseño de la base de datos se ha utilizado la herramienta case ER-Win 3.5.2 (No proporcionada junto con el proyecto por no ser de libre distribución).

El modelo generado por dicha herramienta se encuentra disponible entre la documentación presentada en el directorio [.\Database\Model] → [\[Modelo Erwin\]](#)



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.
Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : **Memoria del proyecto e-Barter**
Descripción : Manual de usuario y especificaciones técnicas del programa de administración.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 4: **PROGRAMA DE ADMINISTRACIÓN** Pagina : 58/133



SECCION 4 - PROGRAMA DE ADMINISTRACIÓN. (Manual de Usuario).



1 Introducción.

Este documento constituye el manual de utilización del programa de administración suministrado en el proyecto eBarter.

A través de esta aplicación se tienen acceso a toda la información gestionada por el sistema de una forma globalizada, permitiendo en todo momento ver y modificar los datos incluidos en el mismo.

Se obtiene una visión de todos los usuarios dados de alta y sus datos relacionados, incluyendo sus recursos y todos los intercambios gestionados (incluso los que para el usuario ya no son visibles por estar en estado de borrado). Esta misma visión y gestión se tiene a nivel de intercambio, estando ambas entidades ligadas en todo momento.

A través de la aplicación se pueden realizar todas las operaciones disponibles para el usuario final, pero además otras propias del administrador como son la gestión de recursos del sistema, el lanzamiento y control del motor de intercambio y la manipulación completa de la base de datos a través de diferentes herramientas proporcionadas como por ejemplo un editor SQL o un gestor de tablas.

Para la utilización de esta herramienta se deben dar de alta en el sistema un nuevo usuario administrador sobre la base de datos EBARTER, así como el registro de dicho usuario en la tabla U_USER, en la cual se definen propiedades de personalización como el tipo de menús a usar o el lenguaje en que debe funcionar la aplicación para ese usuario.

NOTA: En la instalación inicial y arranque del sistema se proporciona el usuario **ebadmin**, con idéntico login como usuario administrador (Estos datos son cargados en las operaciones de inicialización de datos al montar la base de datos sobre la que opera E-Barter).

La aplicación se encuentra estructurada en tres grandes grupos

1. **GENERAL**

- a. Acerca de..
- b. Help
- c. Configuración de la aplicación
- d. Información de conexión
- e. SALIR

2. **DB MANAGEMENT**

- a. Gestión de RECURSOS
- b. Gestión de USUARIOS
- c. Gestión de INTERCAMBIOS
- d. Manipulación de la Base de datos (Editor de tablas)
- e. Manipulación de la Base de datos (Editor SQL)
- f. Generador de datos aleatorios.

3. **eBARTER ENGINEE**

- a. Conexión con cliente WEB
- b. Lanzamiento del motor eBarter
- c. Gestión y visualización de LOG de operaciones.

En las siguientes paginas encontrara un manual de usuario donde se describen las diferentes pantallas disponibles en la aplicación y las operaciones que se pueden realizar en las misma.



2 Acceso y Configuración de la aplicación.

2.1 Acceso a la aplicación

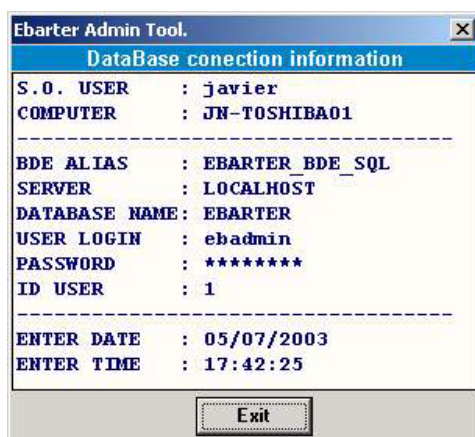
Para iniciar la aplicación ejecute el fichero **eBarterADM.exe** que se encuentra en el directorio raíz de la instalación realizada.

Como paso previo para entrar en la aplicación usted debe autenticarse mediante un login y password valido, para ello debe introducir dichos datos en la siguiente pantalla.



- (Dispone de tres intentos antes de que la aplicación pare su ejecución).
- Login y clave inicial ebadmin / ebadmin

Una vez conectado se le proporciona información sobre la conexión que le permite comprobar si la misma a sido realizada correctamente. Esta información se presenta en la siguiente pantalla.



Tras esto se entra en la aplicación en la pantalla de "acerca de..." donde se tiene información general sobre el proyecto, fecha de realización y sus autores.



2.2 Descripción de zonas y Configuración de la aplicación.

En la siguiente figura puede ver la pantalla correspondiente al "ACERCA DE..." en la que se indican las diferentes áreas en las que se divide la pantalla de la aplicación.

MENU PRINCIPAL



Menu Lateral
(Seleccionable formato
ARBOL o BARRA)

Información de Conexión

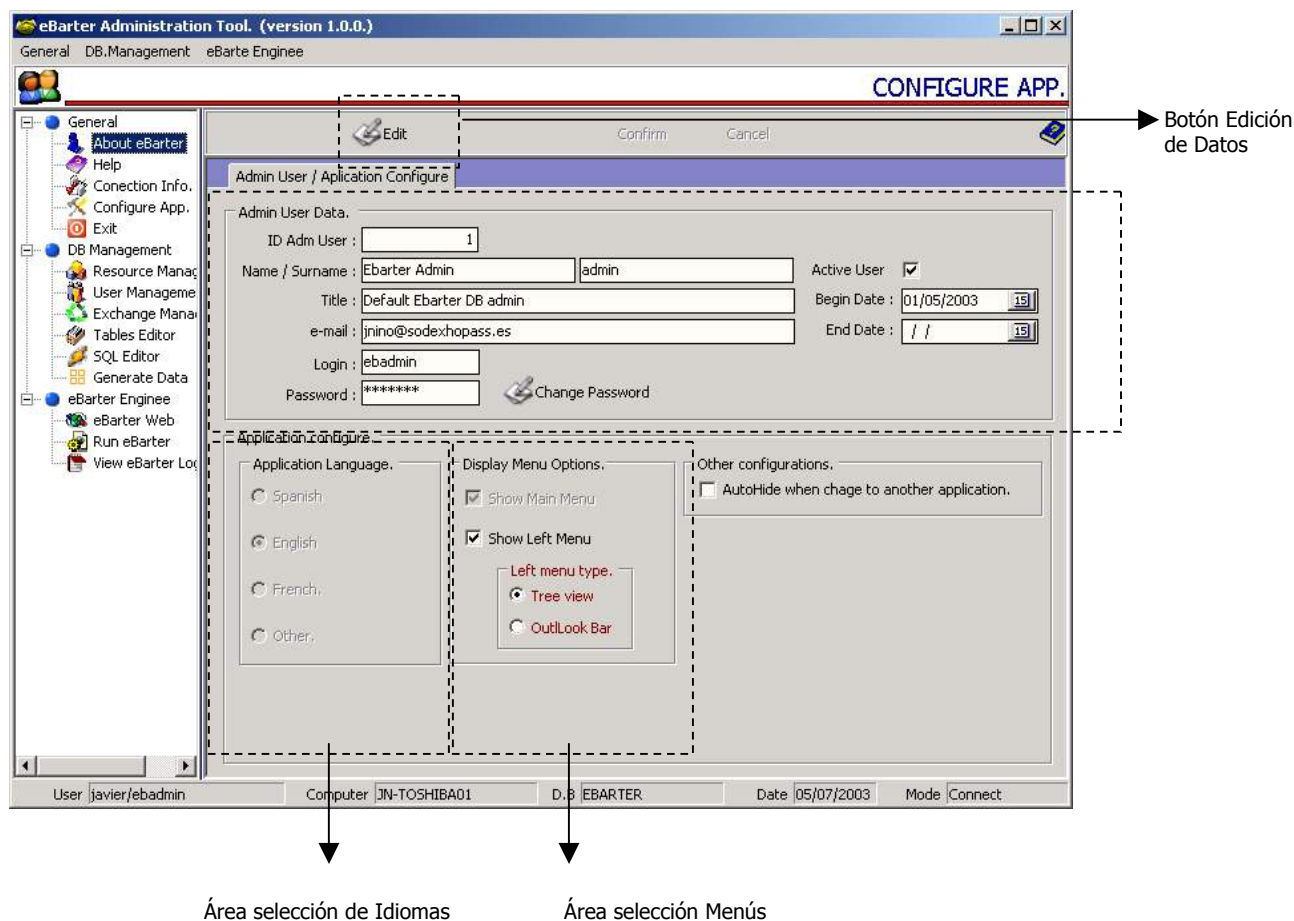
Area de trabajo. (Incrustación de pantallas de
búsqueda y mantenimiento)

A través del menú mostrado en la figura anterior se tiene acceso a las opciones de:

- Ayuda
- Información de Conexión (vista en el apartado anterior)
- Configuración de la aplicación (se detalla a continuación)
- SALIR y cerrar la aplicación.



Se dispone de la posibilidad de cambiar la configuración de la aplicación en los aspectos del idioma utilizado en la interfaz y el tipo de menú utilizado. Para ello seleccione la opción **"Configure App."** del menú General y accederá a la siguiente pantalla.



A través de la opción EDIT podrá cambiar los datos del usuario y la configuración deseada para la aplicación. (En la imagen anterior se ha utilizado la visualización del menú lateral izquierdo en forma de árbol).



3 Gestión de Recursos.

A través de esta opción podemos realizar el mantenimiento de los recursos y categorías de recursos que están disponibles en el sistema para la selección por parte de los usuarios. (Si un recurso requerido no esta disponible el usuario manda una petición a través de la pagina web que el administrador incorpora al sistema en caso que lo considere oportuno).

Para acceder a esta pantalla seleccione la opción RESOURCES del menú DB MANAGEMENT.

Botones de Opción sobre CATEGORÍAS
NUEVA – EDITAR – BARRA DE NAVEGACIÓN – CONFIRMAR O

Opción RESOURCES

Lista de categorías

Category	Level
FOOD	0
DRINK	1
SOFT DRINKS	2
ALCOHOL DRINKS	2
FRUIT	1
FISH	1
MEAT	1
OBJECT	0
COMPUTER	1
BOOK	1

Datos de categoría seleccionada

Category ID.: 1000
Category Name: FOOD
Mother Category:
Category Level: 0

Opciones de Edición y mantenimiento sobre RECURSOS

Lista de recursos en Categoría.
(Relacion Maestro/Detalle con lista de categorías)

ID Resource	Resource Name	System Value

El establecimiento de una relación Maestro-Detalle entre las categorías y los recursos incluidos en ellas se aprecia de forma que en el área de recursos solo se muestran los que están incluidos en la categoría seleccionada.

Recuerde que cuando se edita una categoría se puede establecer una jerarquía a través del campo nivel, pudiendo agrupar los conceptos incluidos en el sistema.

Los botones de mantenimiento se activan o desactivan automáticamente dependiendo de la posibilidad de realizar la acción implementada en ellos (ejemplo. Podremos editar un recurso siempre que exista al menos uno).



4 Gestión de Usuarios.

A través de la gestión de usuarios podremos acceder a la información relacionada con todos los usuarios registrados en el sistema, ya estén activos o dados de baja.

Toda gestión se inicia con una búsqueda a la que se le pueden aplicar un conjunto de filtros AND para después acceder a la pantalla de datos y mantenimiento, en donde se pueden crear, editar o eliminar usuarios registrados. Así mismo desde la pantalla de mantenimiento podremos acceder a toda la información asociada al usuario como son sus recursos y todos sus intercambios asociados.

En los siguientes apartados se presentan las pantallas de gestión de usuarios.

4.1 Búsqueda

A través de esta pantalla se puede realizar una búsqueda de los usuarios del sistema con un amplio conjunto de filtros de selección.

The diagram illustrates the user management interface, specifically the search functionality. It shows two screenshots of the 'eBarter Administration Tool (version 1.0.0.)' with annotations.

Top Screenshot: Search Criteria

The top screenshot shows the 'USER MANAGEMENT' window. A dashed box labeled 'CRITERIOS DE SELECCION' highlights the search filters on the right side of the window. These filters include:

- User ID: (dropdown)
- User Name: (text input)
- User Surname: (text input)
- Login: (text input)
- User Status: (dropdown)
- Registered Date: (dropdown)

A box labeled 'Botones de operación (Su disponibilidad depende de la existencia de datos)' points to the 'New', 'Select', and 'Search' buttons at the top of the window.

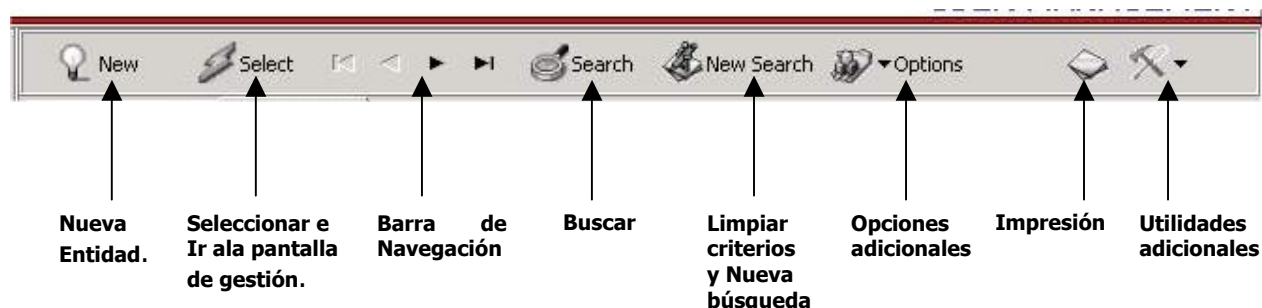
Bottom Screenshot: Search Results

The bottom screenshot shows the same window after a search has been performed. A dashed box labeled 'RESULTADOS DE LA BÚSQUEDA REALIZADA' highlights the search results table. The table has the following columns: IDUSER, LOGIN, NAME, SURNAME, and STATUS. The results are as follows:

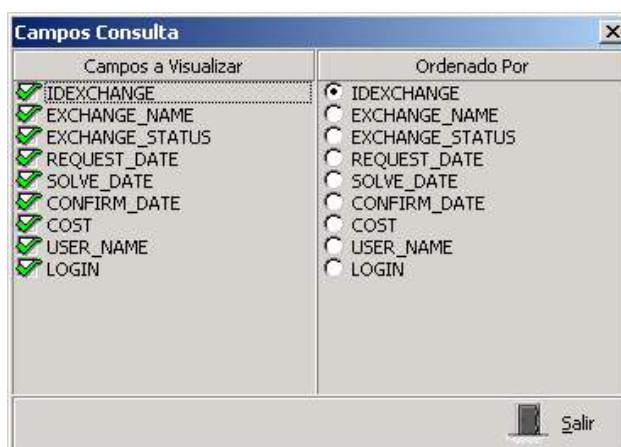
IDUSER	LOGIN	NAME	SURNAME	STATUS
1	user1	Name1	Surname1	ACTIVE
2	USER2	2	2	ACTIVE
3	user3	3	3	ACTIVE
4	RUBEN	RUBEN	RUBEN	ACTIVE
5	RENE	RENE	RENE	ACTIVE
6	javi	JAVIER	NIÑO	ACTIVE
12	yo	yo	yo	ACTIVE
13	tu	tu	tu	ACTIVE
14	el	el	el	ACTIVE



Los botones de las pantallas de búsqueda son todos equivalentes. En la siguiente imagen se detallan sus funcionalidades.



Adicionalmente en toda pantalla de búsqueda se dispone de la posibilidad de ordenar los registros encontrados y filtrar la información mostrada. Para ello pulse el botón derecho del ratón sobre la rejilla de datos y seleccione la opción de **"Hide/Show Column"**, aparecerá una pantalla como la siguiente.



Marque las columnas a visualizar y seleccione el orden que le interese, al pulsar salir volvera a la pantalla de búsqueda y los datos se refrescan según lo indicado en esta ultima pantalla.



4.2 Mantenimiento

Al acceder a la pantalla de mantenimiento usted podrá gestionar toda la información relacionada con el usuario. Las opciones disponibles se encuentran en la parte superior, mientras que podremos acceder al resto de información relacionada cambiándonos entre las pestañas que se encuentran en la parte inferior (manteniendo siempre visible la información del usuario al que hace referencia).

OPCIONES DISPONIBLES PARA EL USUARIO

DATOS DEL USUARIO.

The screenshot shows the 'FUserManagement' application window. It has a menu bar with 'New', 'Edit', 'Confirm', 'Cancel', and 'Delete'. Below the menu bar, there are several input fields for user data: 'Int. Number' (1), 'Status' (ACTIVE), 'Registered Date' (29/06/2003), 'Name' (Name1), 'Surname' (Surname1), 'Address' (Address1), 'Phone 1' (911.11.11.11), 'Phone 2' (empty), 'Country' (SPAIN), and 'Neighborhood' (MADRID). Below these fields, there are tabs for 'User Access Data', 'Resources', and 'Exchanges'. The 'User Access Data' tab is selected, showing fields for 'Login' (User1), 'Password' (1), 'Question' (n), and 'Answer' (n).

The screenshot shows the 'FUserManagement' application window with the 'Resources' tab selected. It displays a table of resources for the selected user. The table has columns for 'Category', 'ID#', 'Resource', and 'Amount'. The data is as follows:

Category	ID#	Resource	Amount
ALCOHOL DRINKS	1001	VODKA	49
ALCOHOL DRINKS	1002	WHISKY	8
FRUIT	1204	BANANAS	2
SOFT DRINKS	1003	COKE	1

Below the table, there are 'Add' and 'Remove' buttons.

RECURSOS DEL USUARIO SELECCIONADO

MANIPULACIÓN DE RECURSOS DEL USUARIO



Si desea modificar los recursos disponibles del usuario utilice los botones de manipulación de recursos del usuario que se muestran en la pantalla mostrada anteriormente.

Si selecciona la opción de añadir recurso se mostrar una pantalla con todos los recursos del sistema en la que se indica el recurso que se quiere añadir y la cantidad del mismo. Si el recurso ya esta disponible para el usuario las cantidades actual y a insertar se suman. A continuación se muestra la pantalla de inserción de recursos para el usuario seleccionado.

ID Resource	Resource Name	Amount	Category
1204	BANANAS	2	FRUIT
1003	COKE	1	SOFT DRINKS
1001	VODKA	49	ALCOHOL DRINKS
1002	WHISKY	8	ALCOHOL DRINKS

Resource Selected : BANANAS
Amount to Give : 0000,00

En la tercera pestaña accedemos a los intercambios asociados al usuario (y recursos incluidos en los mismos), pudiendo realizar sobre ellos cualquier operación que este disponible o bien ir a la pantalla de gestión específica de intercambios.

Las opciones disponibles (siempre dependiendo del estado del intercambio seleccionado son:

1. Ver intercambio → Accede a la pantalla de intercambios con la información para el intercambio seleccionado
2. Añadir un nuevo intercambio (va a la pantalla de intercambios en modo inserción con el usuario prefijado al usuario actual).
3. Editar intercambio. (Va a la pantalla de mantenimiento de intercambios en modo edición con el intercambio actual fijado).
4. Borrar intercambio. (Marca el intercambio como borrado. Estado DELETE).
5. Aceptar Transacción. Acepta la transacción propuesta para el intercambio dado
6. Descartar Transacción. Rechaza la transacción para el intercambio dado.
7. Ver transacción. Va a la pantalla de visualización de datos de la transacción (mostrada con posterioridad en este mismo documento).



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.
Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : **Memoria del proyecto e-Barter**
Descripción : Manual de usuario y especificaciones técnicas del programa de administración.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 4: **PROGRAMA DE ADMINISTRACIÓN** Pagina : 68/133

FUserManagment

New Edit < > Confirm Cancel Delete

Int. Number Status Registered Date

Name

Surname

Address

Phone 1 Phone 2

e-mail

Country Neighborhood

User Access Data | Resources | Exchanges

Exchange List.

ID#	Register Date	Name	Status	Solve Date.	Confirm Date
44	02/07/2003	USER 1 (2C > 2 TR -- 43)	DELETED		
45	03/07/2003	u1_e->@1t + 1w = 2v / 1v = EBARTER		03/07/2003	
46	03/07/2003	u1_e->@1t + 1w = 2v / 1v = DISCARDED BY OR			
47	03/07/2003	u1_e-> 2w + 1c = 2b	ACTIVE		

Resource to Give.

Resource	Amount
TRINA	1
WHISKY	1

Resources to Obtain.

Resource	Amount
VODKA	2

View Exchange
Add
Edit
Remove
Accept
Discard
View Transaction

DATOS DE INTERCAMBIOS ASOCIADOS AL USUARIO

**OPCIONES DISPONIBLES
PARA EL INTERCAMBIO
SELECCIONADO**



5 Gestión de Intercambios.

De funcionalidad equivalente a la administración de usuarios pero para intercambios. También se encuentra relacionada con la pantalla de usuarios permitiéndonos ver y manipular los datos de intercambios y usuarios comunicándonos entre las dos pantallas.

5.1 Búsqueda

Las pantallas de búsqueda para intercambios tienen idénticas funciones e interfaz que la búsqueda de usuarios. Las pantallas específicas se muestran en las siguientes imágenes.

eBarter Administration Tool. (version 1.0.0.)
General DB.Management eBarter Engine

EXCHANGES

Search Approaches

Exchanges Data:

Exchange ID:
Exchange Name:
Exchange Status:
Registered Date:
Solve Date:
Confirm Date:

User Data:

User ID:
User Name:
User Surname:
Login:
User Status:
Registered Date:

User |javier/ebadmin Computer |JN-TOSHIBA01 D.B |EBARTER Date |05/07/2003 Mode |Connect

eBarter Administration Tool. (version 1.0.0.)
General DB.Management eBarter Engine

EXCHANGES

Search Approaches Search Result

Number of registers find: 56

INDEX	EXCHANGE	EXCHANGE_NAME	EXCHANGE_STATUS	REQUEST_DATE
36	USER 1	(1T + 1V > 2W)	DELETED	01/07/2003 21:1
37	1 W + 2 T x 6 B		EBARTER SUGGESTION	02/07/2003
38	1 W + 2 T x 6 B / 4 C x 4 O + 3 P		DISCARDED BY OR RELATION	02/07/2003
39	6 V x 12 F		ACTIVE	02/07/2003
40	2 W x 1 C + 4 T + 6 O4		ACTIVE	02/07/2003
41	21 B x 1 w + 1 t		ACTIVE	02/07/2003
42	5 b x 1 w 1 t		EBARTER SUGGESTION	02/07/2003
43	USER 1	(1C + 1T x 2F // 2C x 2F)	DELETED	02/07/2003 21:1
44	USER 1	(2C > 2 TR -- 43)	DELETED	02/07/2003 21:1
45	u1_e->@1t + 1w = 2v / 1v = 3c + 1w		EBARTER SUGGESTION	03/07/2003
46	u1_e->@1t + 1w = 2v / 1v = 3c + 1w		DISCARDED BY OR RELATION	03/07/2003
47	u1_e-> 2w + 1c = 2b		ACTIVE	03/07/2003
48	u2_e-> 1b = 1c+1f		ACTIVE	03/07/2003
49	u2_e-> @1b = 1w / 1b = 1v		ACTIVE	03/07/2003
50	u2_e-> 1b = 1w / @1b = 1v		ACTIVE	03/07/2003
51	u3_e-> 2v = 1t+1w		EBARTER SUGGESTION	03/07/2003
52	1Pera por 90Whiskies		DONE	03/07/2003
53	1vodka por 12filetes		DELETED	03/07/2003
54	pepe		ACTIVE	03/07/2003
55	pepe432		ACTIVE	03/07/2003
56	93 whisky + 1 Coke por 1 pera		DONE	03/07/2003

User |javier/ebadmin Computer |JN-TOSHIBA01 D.B |EBARTER Date |05/07/2003 Mode |Connect



5.2 Mantenimiento

A través de la pantalla de mantenimiento de intercambios tenemos acceso a la manipulación completa de los mismos.

Desde esta pantalla podremos :

1. Insertar, Editar y eliminar intercambios.
2. Ver si hay intercambios relacionados.
3. Ver la información del usuario al que pertenece y acceder a la pantalla de mantenimiento de usuarios para la gestión del mismo
4. Aceptar las transacciones propuestas por el sistema. Marca el intercambio como **WAIT TO CONFIRM** a la espera de que todos los intercambios relacionados sean confirmados. Cuando esto sucede automáticamente el sistema marca el intercambio como **DONE**. En caso de que un intercambio sea rechazado se marcará como **DISCARD BY OTHER USER**
5. Rechazar las transacciones propuestas por el sistema. Descarta el intercambio pasándolo a activo o borrado según selección del usuario, el resto de intercambios relacionados se marcan como **DISCARD BY OTHER USER**.
6. Ir a la pantalla de detalles de transacciones. (Evaluación de la transacción)

Los puntos 4 y 5 solo son accesibles si el intercambio seleccionado esta en estado de EBARTER_SUGESTION o WAIT TO CONFIRM.

La pantalla de mantenimiento de intercambios se presenta en la siguiente imagen.

BOTONES DE ACCIONES DISPONIBLES SOBRE EL INTERCAMBIO

DATOS DEL INTERCAMBIO

DATOS DE LOS RECURSOS INCLUIDOS EN EL INTERCAMBIO

Botones para Añadir o eliminar recursos a dar

Botones para añadir o eliminar recursos a obtener



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Manual de usuario y especificaciones técnicas del programa de administración.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 4: PROGRAMA DE ADMINISTRACIÓN

Página : 71/133

Si el intercambio tiene otros relacionados aparecerá una pestaña indicándolo y dentro podremos ver los detalles de dichos intercambios.

ID#	Register Date	Name	Status	Solve Date	Confirm Date	Cost
46	03/07/2003	u1_e->@1t + 1w = 2v / 1v = 3c + 1w	DISCARDED BY OR			

INTERCAMBIOS RELACIONADOS.

En la tercera pestaña podremos ver los datos del usuario al que pertenece el intercambio e incluso acceder a la pantalla de mantenimiento de usuarios con sus datos prefijados.

Int. Number	Status	Registered Date
1	ACTIVE	29/06/2003

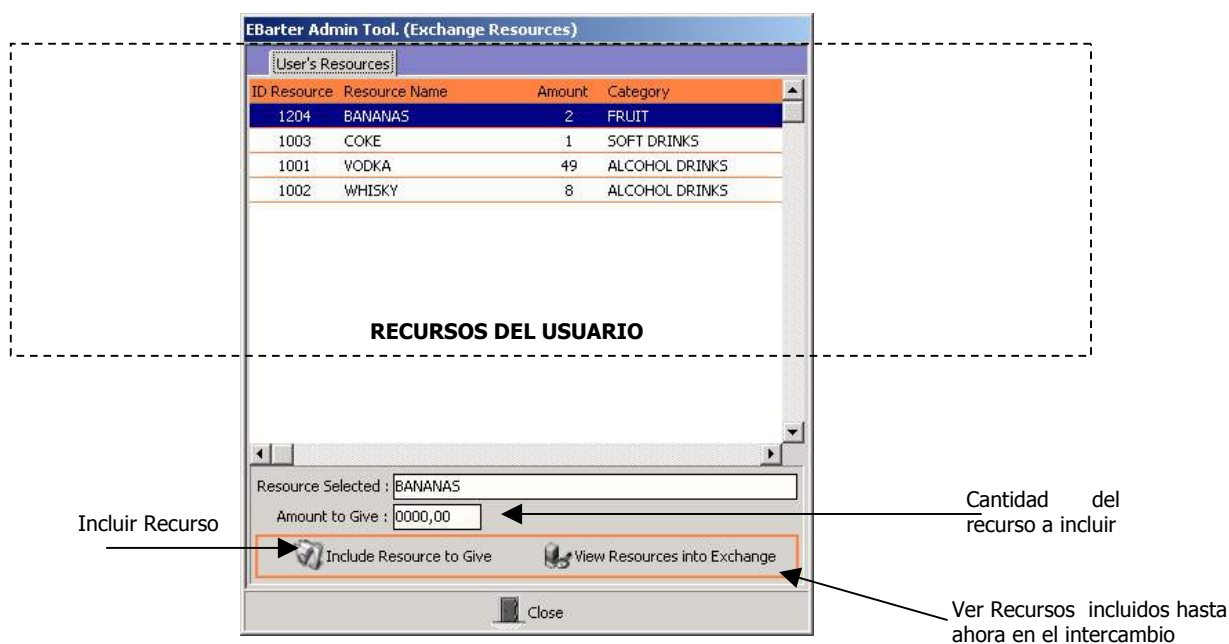
Name	Surname	Address	Phone 1	Phone 2	Country	Neighborhood
Name1	Surname1	Address1	911.11.11.11		SPAIN	MADRID



5.2.1 Edición de un nuevo intercambio.

Para añadir un nuevo intercambio seguiremos los siguientes pasos:

1. Pulsar sobre el botón añadir.
2. Introducir el nombre del intercambio (los datos de numero de intercambio, estado y fecha son proporcionados por el sistema y no manipulables para asegurar la integridad de los datos y procesos internos).
3. Seleccionar el usuario. Introduzca numero de identificación y pulse <Enter> o bien utilice el botón de localización de usuario.
4. Seleccione los recursos a dar. Pulse el botón indicado en la pantalla de mantenimiento de mensajes. Se mostrar una pantalla como la que se muestra a continuación en la que podrá seleccionar los recursos y cantidades que se desean dar atendiendo a la mochila del usuario.



Observaciones:

- a. Para poder ir a esta pantalla se debe haber seleccionado previamente el usuario.
 - b. Una vez incluidos recursos del usuario este no puede ser cambiado. Si quiere cambiar de usuario necesita la edición del intercambio y crear uno nuevo.
 - c. La cantidad del recurso a dar no puede sobrepasar la que el usuario tenga disponible en su mochila.
5. Seleccione los recursos a obtener.
Una vez incluidos los recursos que se están dispuestos a dar, realizamos la misma operación para indicar los recursos que se quieren obtener. De forma equivalente usamos el botón de insertar recursos a obtener y pasaremos a una pantalla equivalente a la que se muestra a continuación.



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Manual de usuario y especificaciones técnicas del programa de administración.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 4: PROGRAMA DE ADMINISTRACIÓN Pagina : 73/133

ID Resource	Resource Name	Category	System Value
1203	APPLES	FRUIT	3
1204	BANANAS	FRUIT	2
1401	BEFF	MEAT	12
1003	COKE	SOFT DRINKS	0,6
5101	COMPUTER P4	COMPUTER	1000
4101	DICCIONARIO DE INGLES	BOOK	5
1004	FANTA	SOFT DRINKS	0,6
1205	MELONS	FRUIT	2
1202	ORANGES	FRUIT	3
1201	PEARS	FRUIT	2
5102	PRINTER	COMPUTER	300
1006	RUM	ALCOHOL DRINKS	12
1400	STEAK	MEAT	10
1005	TRINA	SOFT DRINKS	0,6

Resource Selected : APPLES
Amount to Obtain : 0000,00

Include Resource to Obtain View Resources into Exchange

Close

En cualquiera de las pantallas de inserción de recursos podemos ir a visualizar el estado global de todos los recursos incluidos en el intercambio pulsando el botón de **"View Resources into Exchange"**, que nos lleva a la siguiente pantalla:

View of resources included into exchange.

Resource to Give.

Resource	Amount
BANANAS	4
VODKA	10

Resources to Obtain.

Resource	Amount
ORANGES	5
WHISKY	5

Close



6. Grabar el intercambio.

En este caso debe seleccionar si grabar el intercambio solo o desea relacionarlo con otro a meter a continuación. Por ello al pulsar sobre el botón grabar deberá seleccionar una de esta opciones sobre la siguiente pantalla.

Ebarter Admin Tool. (Select Exchange Save Mode).

Save Mode.

☒ Save Exchange ☐ Continue with Conditional Exchange

Close

Si selecciona la opción de "**Continue with conditional exchange**", el intercambio no sera grabado y quedara a la espera de una nueva inserción. Cuando finalmente selecciona la opcion "**Save Exchange**" todos los intercambios introducidos son grabados en el sistema de forma transaccional. (los intercambio relacionados OR se gestionan en conjunto).

5.3 Pantalla de detalles de transacción.

Si el intercambio tiene detalles de transacción (El sistema a dado una propuesta para el mismo) podemos acceder a sus detalles a través del botón "**view transaction**" que nos lleva a la siguiente pantalla.

Ebarter Admin Tool. (View Transaction Detail)

Transaction Details.

Exchange Data.

Int. Number: 45 Status: EBARTER SUGGESTION Registered Date: 03/07/2003 Solved Date: 03/07/2003 Confirm Date: / /

Name: u1_e->@1t+1w=2v/1v=3c+1w

Transaction Cost: 1,2

User Data login: user1 / ID#: 1 / Name: Name1,Surname1

Resources to Give:

Resource	Amount	Target Exch.	Target User
WHISKY	1	51	user3
TRINA	1	51	user3

Resources to Obtain:

Resource	Amount	Source Exch.	Source User
VODKA	2	51	user3

Exchange ID: 51 Status: EBARTER SUGGESTION

Exch. Name: u3_e-> 2v = 1t+1w

User Name: 3,3 Address: 3 e-mail: 3 Country: USA Neighborhood: MIAMI

Exchange ID: 51 Status: EBARTER SUGGESTION

Exch. Name: u3_e-> 2v = 1t+1w

User Name: 3,3 Address: 3 e-mail: 3 Country: USA Neighborhood: MIAMI

Accept Transaction Discard Transaction Close

DATOS DEL INTERCAMBIO DE REFERENCIA DESDE EL QUE SE ACCEDE

RECURSOS A DAR (A QUE INTERCAMBIO Y USUARIO SE LES DA)

RECURSOS OBTENIDOS (DE QUE INTERCAMBIO Y USUARIO SE OBTIENEN)

ACEPTACIÓN O RECHAZO DE LA TRANSACCIÓN



6 Manipulación de la Base de Datos.

Se dispone de un conjunto de herramientas para la manipulación y edición de las tablas incluidas en la base de datos sin la utilización de las herramientas proporcionadas por SQL-Server.

El objetivo de estas utilidades es poder manipular todo el sistema de datos desde la propia aplicación de administración del sistema

6.1 Edición de Tablas.

Pantalla en la que podemos seleccionar las diferentes tablas de datos y proceder a su manipulación.

Botones de Mantenimiento de la tabla

Tablas del sistema

Datos incluidos en la tabla seleccionada

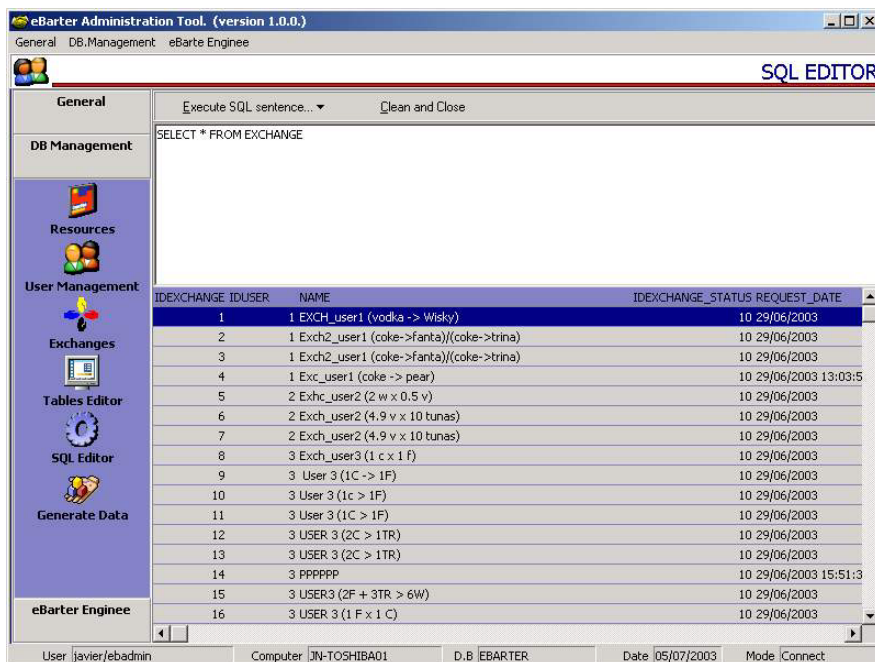
6.2 Editor de sentencias SQL.

Se dispone también de una pantalla de edición y ejecución de sentencias SQL, ya sean de selección y/o mantenimiento (inserción, update y delete).

Para su utilización simplemente tiene que escribir la sentencia SQL que desee, selecciónela y pulse sobre el botón de ejecutar o la tecla de función F5.

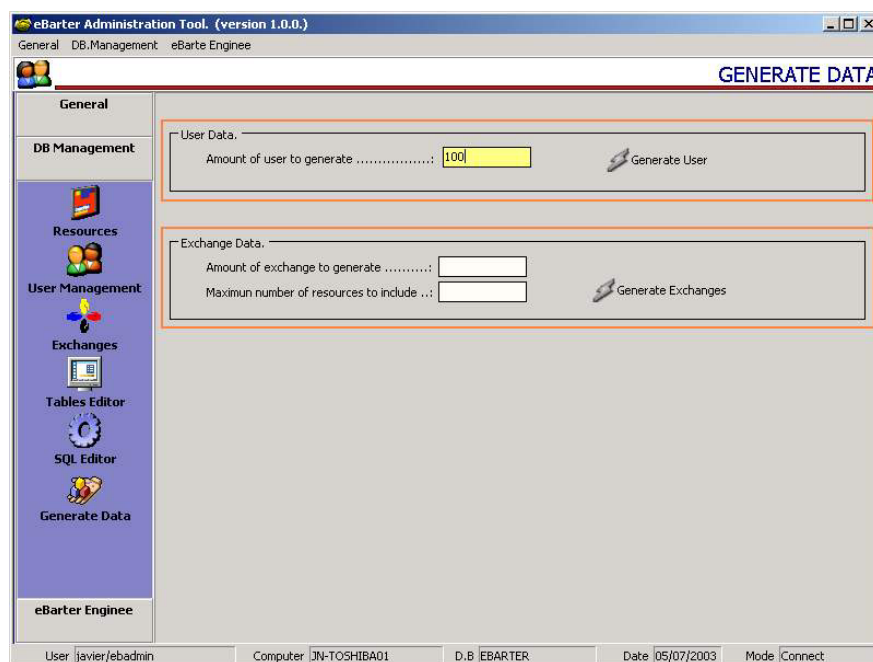


La pantalla de edición de sentencias SQL es la siguiente:



6.3 Generador de Datos aleatorios.

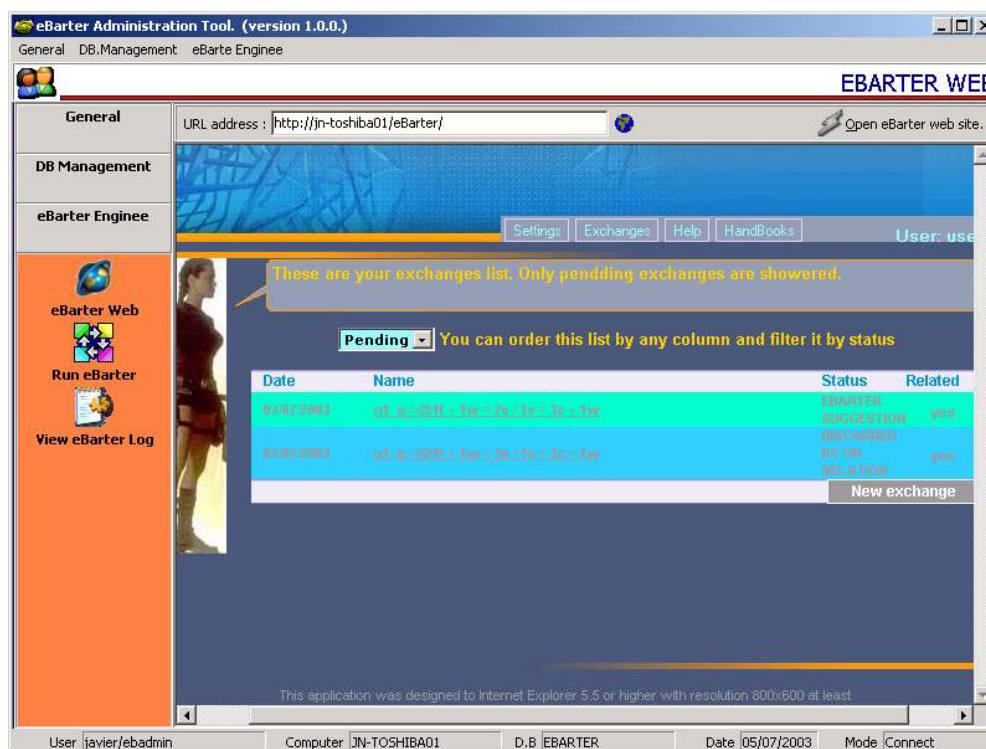
Para cargar el sistema de datos para su evaluación se ha preparado una herramienta de carga a la cual se tiene acceso desde la aplicación de administración a través de la opción "**GENERATE DATA**" del menú "BD MANAGEMENT"





7 Enlace con el cliente WEB

En la sección de "EBARTER ENGINEE" podemos encontrar en primer lugar un enlace con el cliente WEB utilizado por los usuarios del sistema, (opción "**eBarter Web**"), permitiéndonos comprobar su funcionamiento y realizando todas las operaciones que están disponibles en el mismo.





8 Ejecución del motor de transacciones.

A través de esta opción (**RUN eBARTER**) el administrador puede lanzar el motor de intercambios, núcleo del sistema que realiza los emparejamientos y propone las posibles transacciones a realizar para cada uno de los intercambios resuelto. Este modulo se encuentra en la carpeta JAVA del proyecto y es invocado a través de la presente pantalla, aunque se puede realizar su ejecución de forma independiente.

Para tener una visión clara de los intercambios que va a gestionar el motor en la ejecución, en esta pantalla se presentan todos los elementos que van a ser procesados, estructurados desde el punto de vista de los intercambios y del usuarios (permite al administrador hacer las comprobaciones previas que considere oportuna). Así mismo , una vez lanzado el motor se puede refrescar esta pantalla y ver fácilmente los elementos que han quedado son solución.

En la siguiente imagen se muestra la pantalla de ejecución eBARTER.

eBarter Administration Tool. (version 1.0.0.)

General DB.Management eBarter Engines

RUN EBARTER

View Exchange List to Process View User List to Process

Number of registers find : 9

Register Date	Exchange Name	Status	User Name
02/07/2003	6 V x 12 F	ACTIVE	RENE RENE
02/07/2003	2 W x 1 C + 4 T + 6 O4	ACTIVE	RENE RENE
02/07/2003	21 B x 1 w + 1 t	ACTIVE	RENE RENE
03/07/2003	u1_e-> 2w + 1c = 2b	ACTIVE	Name1 Surname1
03/07/2003	u2_e-> 1b = 1c+1f	ACTIVE	2 2
03/07/2003	u2_e-> @1b = 1w / 1b = 1v	ACTIVE	2 2
03/07/2003	u2_e-> 1b=1w / @1b = 1v	ACTIVE	2 2

Exchange Detail

Exchange Data:

Exchange ID: 39 Status: ACTIVE

Exchange Name: 6 V x 12 F

Registered Date: 02/07/2003

User Data:

User Name: RENE

User Surname: RENE

Login: RENE

Registered Date: 02/07/2003

Resource to Sell:

Resource	Amount
VODKA	6

Resources to Buy:

Resource	Amount
FANTA	12

Boton de ejecución.
2 modalidades

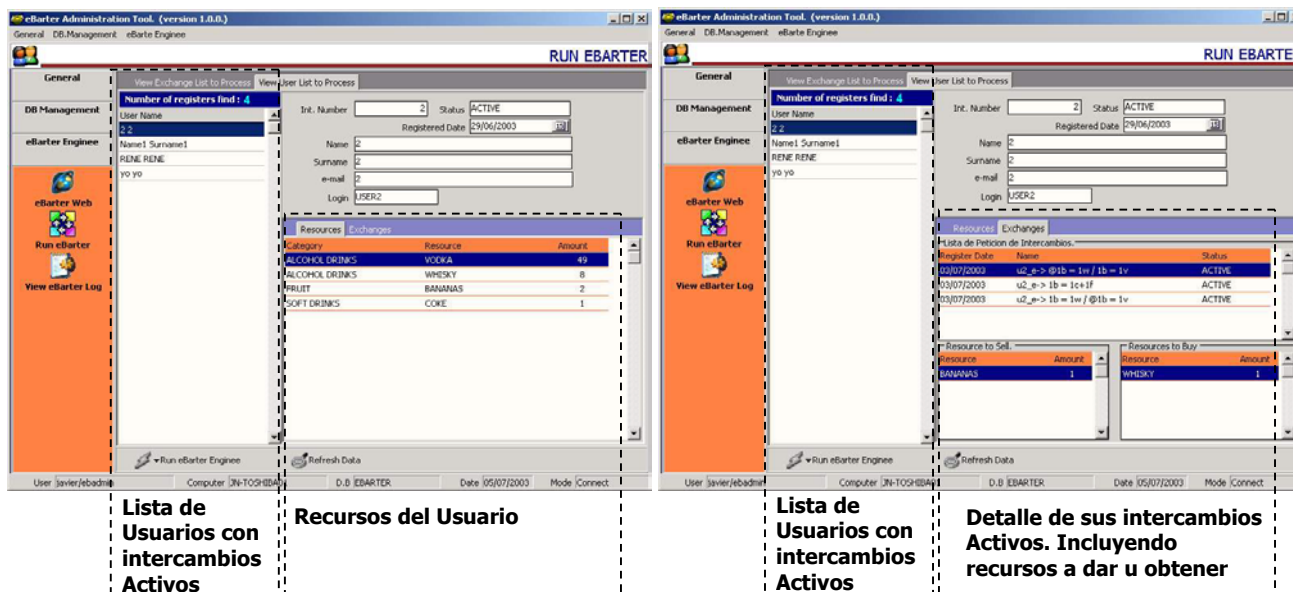
- (Interfaz grafico de control por parte del administrador.
- Ejecución en Batch. (mas rápida y programable como tarea)

Botón para refrescar los datos mostrados .

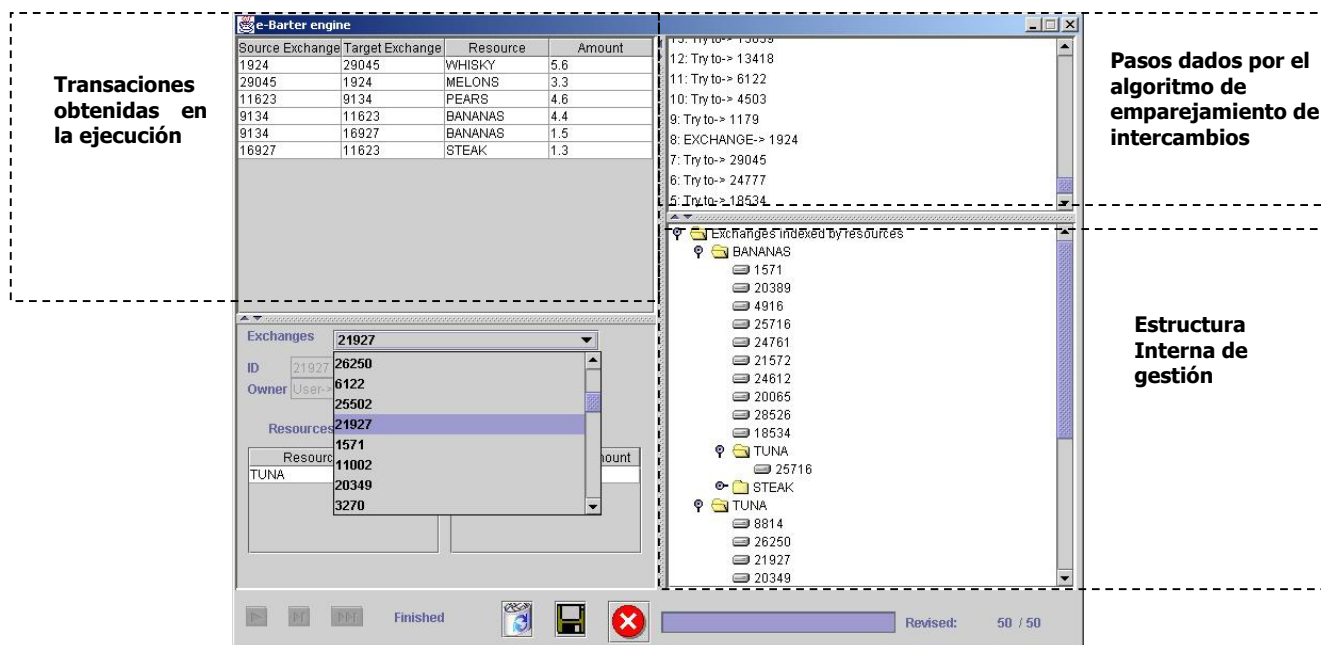
Datos de Intercambios que van a ser procesados.

Detalles de recursos y usuario al que pertenece correspondientes al intercambio seleccionado en la rejilla superior

De forma similar podemos tener esta misma visión pero enfocada a los usuarios que tienen intercambios activos para entrar en el proceso de búsqueda de posibles transacciones.



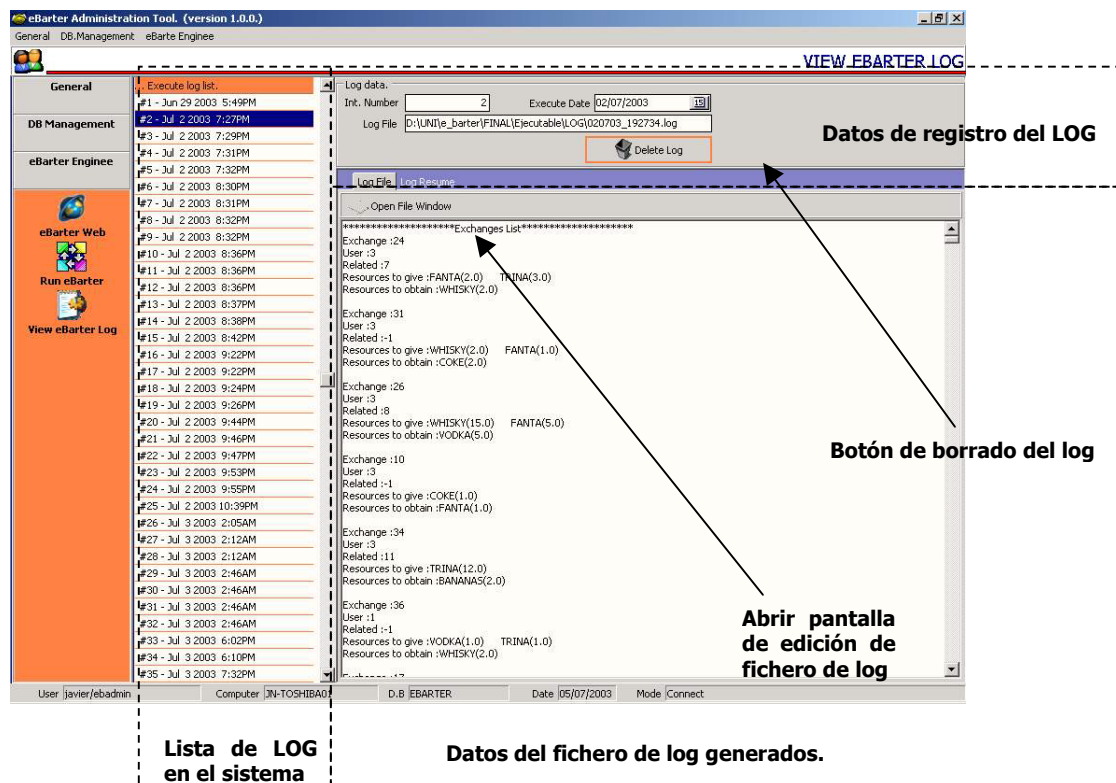
Al Pulsar sobre el boton de ejecución del motor detransacciones se invoca la ejecución del motor JAVA explicado con anterioridad enla sección 2 de este mismo documento. Recuerde que puede optar por la ejecución en modo BATCH o de ADMINITRACION (con interfaz visual pero que consume más recursos del sistema y tiene tiempos de procesamiento peores). A continuación se muestra la pantalla del motor de intercambios en formato gráfico.





9 Fichero de Log.

La ultima opción del programa de administración se corresponde con la gestión de los LOG generados por el motor de intercambios. A través de la opción **"Vive eBarter Log"** podemos ver todos los registros de log generados y borrar aquellos que nos interese, ya que estos ficheros nos ocupan un gran volumen en el disco duro del sistema.



Se dispone de una pantalla de visualización del fichero de log ampliadas con opción de búsqueda de palabras clave, apertura y grabación de ficheros, configuración de impresora, impresión, etc.





10 Requerimientos e Instalación de la aplicación.

1. Para la instalación ejecute el programa que se proporciona para dicho fin.
2. Instalación del BDE ((Borland Database Enginee)).
3. Configuración del BDE para el acceso de la aplicación a la base de datos. Para ello crear una conexión a SQL Server (preferiblemente con los drivers nativos proporcionado, aunque también puede realizarse mediante conexión ODBC). El nombre de esta entrada en el BDE debe de ser :

- Nombre alias : **EBARTER_BDE_SQL**
- Nombre de la base de datos (DATABASE NAME) : **EBARTER**
- Configurar adecuadamente el nombre del HOST y SERVER.

(Consulte la ayuda de esta aplicación si tiene problemas en la configuración y conexión)

4. El sistema requiere una resolución de pantalla mínima de 800x600.



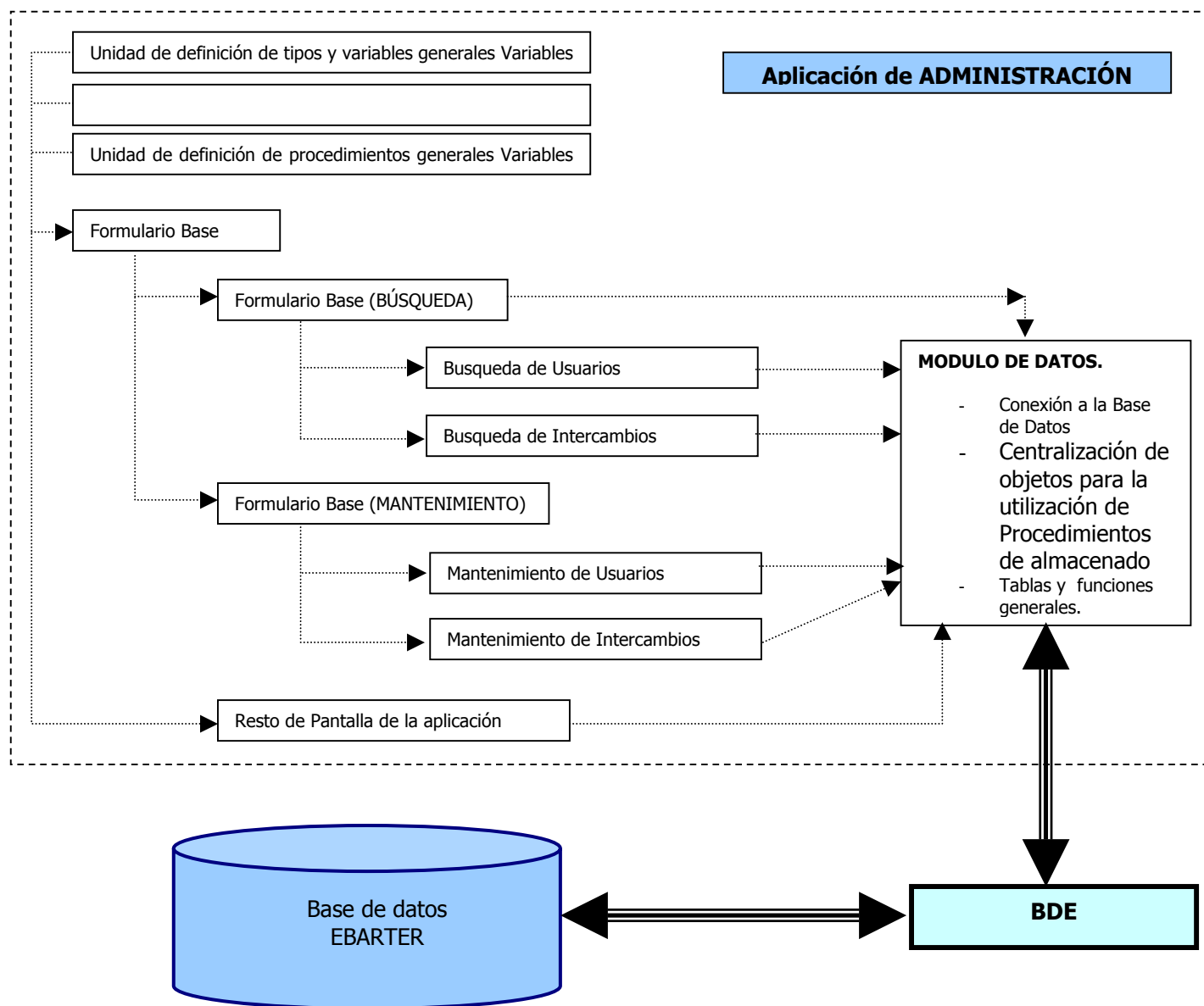
11 Información Adicional.

La aplicación de administración ha sido implementada bajo el entorno de desarrollo **Delphi 7 Enterprise edition**.

Además del conjunto de componentes estándar suministrados por esta herramienta se han utilizado una serie de paquetes y componentes freeware para mejorar el aspecto gráfico de la misma, si bien, dichos componentes no aportan en las labores de procesamiento y gestión implementadas.

En el desarrollo de la aplicación se han usado los conceptos de herencia de la programación orientada a objetos (incluida en el entorno), de tal forma que facilita su mantenimiento y reduce la cantidad de código fuente a desarrollar.

A continuación se presenta un esquema de la estructura modular de la aplicación.





Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: **CLIENTE WEB**

Página : 83/133



SECCION 5 - CLIENTE WEB



1 Web e-Barter

La aplicación web del sistema e-Barter es la fuente e interfaz de información proporcionada al sistema y al usuario.

1.1 Interfaz:

e-Barter es un sistema lucrativo basado en la dependencia del hombre por conseguir elementos mas o menos útiles en su vida diaria. Facilitamos el poder pedir dichos elementos a cambio de dinero (Compra) o a cambio de recursos que el cliente desee intercambiar (Trueque).

Para ello es necesario una interacción del cliente con el sistema e-Barter que se realiza mediante un portal Web en donde el cliente se registra y tiene a su disposición toda la información necesaria para rellenar los formularios que formalizan el intercambio.

1.2 Fuente de información:

e-Barter intentara solucionar todos los intercambios que se encuentren en la base de datos, sin embargo, es necesario previamente que los clientes hayan interactuado con el sitio web para aportar la información con la que el sistema pueda trabajar.

2 Especificación de Requisitos

El sitio Web está implementado mediante páginas dinámicas ASP (Active Server Pages), son páginas no estáticas que se crean online según la interacción del usuario.

Cada página ASP contiene código implementado en Visual Basic y texto HTML la unión de la ejecución del código mediante la interacción del usuario con la página, modifica, define o crea la página web con la que el usuario esta trabajando en ese momento.

Además, lleva incluido código Java para resolver diversas utilidades proporcionadas al usuario en el sistema cliente, a diferencia de ASP que se ejecuta únicamente en el sistema Servidor

- o El código en Visual Basic es interpretado en el servidor y devuelto al usuario.
- o El código Java es interpretado en el cliente y ejecutado con la interacción del usuario.

Para una interpretación correcta del código ASP, es necesario la instalación de Internet Information Server 5 (IIS5) que esta incluida en el Sistema Operativo Windows 2000 y XP Professional.

Una página ASP recibe y almacena casi toda su información en una Base de Datos, en este caso, hemos utilizado SQL Server



2.1 Sistema Operativo:

Hablamos de páginas con código en Visual Basic e IIS5, por lo tanto es obvio que hablamos de Windows.

Se trata de un sistema Web Cliente Servidor.

2.1.1 Servidor:

- El servidor puede estar configurado con Windows 2000 (Preferentemente) o NT. Windows 2000 tiene la ventaja de ser un SO mas estable que NT además de tener instalado IIS5 que a diferencia de NT (IIS4) que debe ser actualizado, pues sino, puede no interpretar correctamente el código ASP.
- Hace falta instalar las librerías de ASPMail tanto en 2000 como NT, son dlls necesarias para el envío automático de e-mail desde el sitio web.
- Para la conexión con la Base de Datos es necesario crear un ODBC con el nombre "ebarterSQLServer" que se relacione con la Base de Datos "EBARTER"

2.1.2 Cliente:

- EL cliente para poder ver y navegar correctamente en el sitio Web, necesita IE5.5 pues los objetos empleados en el código Java del cliente son objetos definidos y creados a partir de la versión 5.5 del browser.
- Por tanto, puede utilizar cualquier sistema operativo Windows con IE5.5



3 Estructura Web e-Barter

En este apartado vamos a revisar la estructura de la web del proyecto e-Barter y veremos que contiene cada archivo del directorio web y para que sirve.

La estructura es la siguiente:

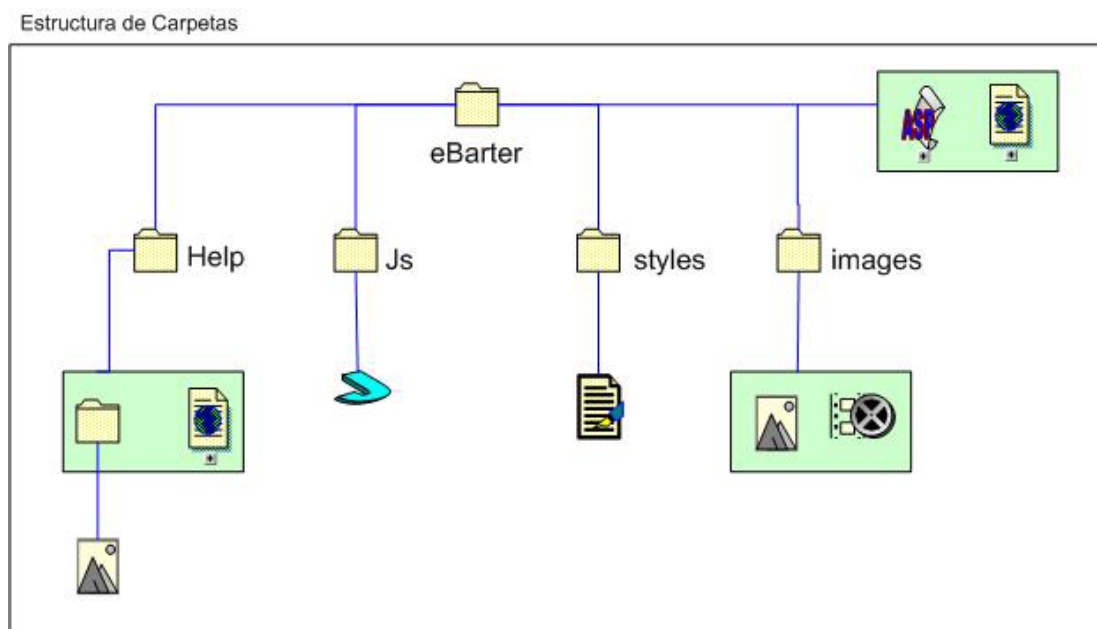
Estructura de directorios	
/	Directorio raíz donde se encuentran todos las páginas .asp
/help	Directorio con todas las páginas de ayuda que el usuario podrá consultar online.
/styles	Este directorio contiene las hojas de estilo comunes a todas (o casi todas las páginas) de la web. Si una página tiene un estilo propio el estilo no se define en este directorio sino dentro de la propia página.
/images	Aquí están todas las imágenes que hay en la web .gif y .jpg
/js	En este directorio se guardan todos los scripts complejos escritos en JScript o JavaScript (pej: menu.js, ventana.js, desplegable.js) o comunes a varias páginas (pej: talker.js).

Un servidor Web se encarga mostrar la página Web al cliente que lo solicite, sin embargo, para ello es necesario configurar adecuadamente la estructura de archivos.

La carpeta principal denominada "eBarter" debe estar guardada en la siguiente dirección física del servidor.

☞ C:\inetpub\wwwroot\

Y no modificar la estructura interna de la carpeta "eBarter"





4 Documentación páginas web

Cada página .asp tiene un algoritmo determinado en función de la tarea que deba desempeñar, sin embargo, podemos agrupar varias de ellas para que en conjunto nos ofrezcan un determinado servicio.

Una página Web, destina a validar información, usuario y contraseña, por si sola no tiene mayor funcionalidad que devolver verdadero si un usuario se ha validado correctamente o falso si ese usuario no existe o se ha olvidado de su contraseña.

Ahora la agrupamos con el formulario de registro, en el caso de devolver falso porque no existe el usuario podemos llamar a la página de registro, o llamar a la página destinada a recordar contraseña en el caso de que el usuario exista y se haya olvidado la contraseña...

La unión de páginas ASP nos da lugar a servicios destinados a que el cliente tenga una navegación sencilla y fácil

Iremos viendo para cada servicio que páginas intervienen y cual es su role en él. Además veremos como están estructuradas internamente y a que tipo de datos tienen acceso.

Lo primero que debemos de ver es una estructura de todas las páginas y su relación entre ellas. Hemos decidido hacer esta división en 4 partes.

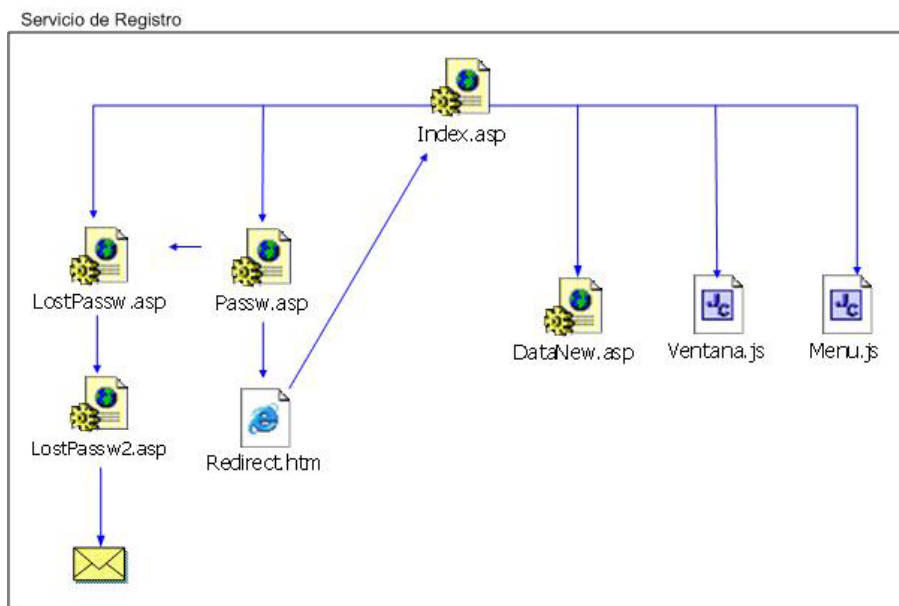
Veremos como la página index.asp es la madre de todas las páginas que veremos en la Web y por tanto de los 4 apartados que desarrollaremos a continuación.

- ⇒ [Servicio de registro.](#)
- ⇒ [Servicio de gestión de intercambios.](#)
- ⇒ [Páginas comunes.](#)
- ⇒ [Servicio de ayuda.](#)



4.1 Servicio de registro

Este servicio representa las distintas funcionalidades que se presentan a un usuario desconectado (excluyendo la ayuda).



En esta parte de la aplicación un usuario podrá registrarse, autenticarse, o solicitar el recordatorio de password. Además hay una página de presentación para llamar la atención del usuario.

Todas estas funcionalidades son alcanzables a través del menú principal ubicado en la página *Index.asp*

4.1.1 Menú principal

Este menú se encuentra en la página *Index.asp* y utiliza los archivos de script *Ventana.js* y *Menú.js* para desplegar y contraer las distintas opciones del menú.





la página inicial de login permite a un usuario del sistema entrar a trabajar en él. El servicio se lleva a cabo en la página *Passwd.asp*. Esta página nos pide los datos y los comprueba con los de la BBDD y si son correctos nos remite a la página *Redirect.htm* que automáticamente nos mete en el sistema haciendo que cambie el menú principal.

En caso de que las credenciales suministradas por el usuario no sean correctas se le muestra un enlace donde podrá solicitar el recordatorio de password.

4.1.2 Recordar password

En esta parte de la aplicación, un usuario que hay olvidado su password puede solicitar que se le recuerde.

Para que un usuario pueda solicitar el recordatorio de password se le hará la pregunta que usuario dio al darse de alta y si la responde correctamente se le enviará el password por e-mail.

Este servicio lo lleva a cabo la página *LostPassw.asp* donde se le pide el login, y *LostPassw2.asp* donde se le formula la pregunta y en caso correcto se le enviará por e-mail.

The screenshot shows a web browser window with a blue header and a yellow navigation bar containing 'File', 'Help', and 'HandBooks' menus. The main content area has a dark blue background with yellow text that reads 'Fill in with your login and you'll receive the password in your mail account'. Below this text is a 'Login' label followed by a light blue input field. At the bottom of the form is a 'Send' button.

The screenshot shows a web browser window with a blue header and a yellow navigation bar containing 'File', 'Help', and 'HandBooks' menus. The main content area has a dark blue background with yellow text that reads 'Answer the question to recovery your password'. Below this text are two labels: 'Question:' and 'Response:', each followed by a light blue input field. At the bottom of the form is a 'Send' button.



4.1.3 Registrarse.

Este servicio permite a un usuario darse de alta en el sistema. El servicio lo lleva a cabo la página *DataNew.asp* pide los datos necesarios al nuevo usuario y posteriormente tras comprobar que no hay errores los guarda en la BBDD.

Fill in this form with your details

Login*	<input type="text"/>	Question	<input type="text"/>
Password*	<input type="text"/>	Response*	<input type="text"/>
re-password*	<input type="text"/>		
<hr/>			
Name*	<input type="text"/>	Phone*	<input type="text"/>
Surname*	<input type="text"/>	Phone2	<input type="text"/>
e-mail*	<input type="text"/>	Country	BELGIUM
Address	<input type="text"/>	Neighborhood*	<input type="text"/>

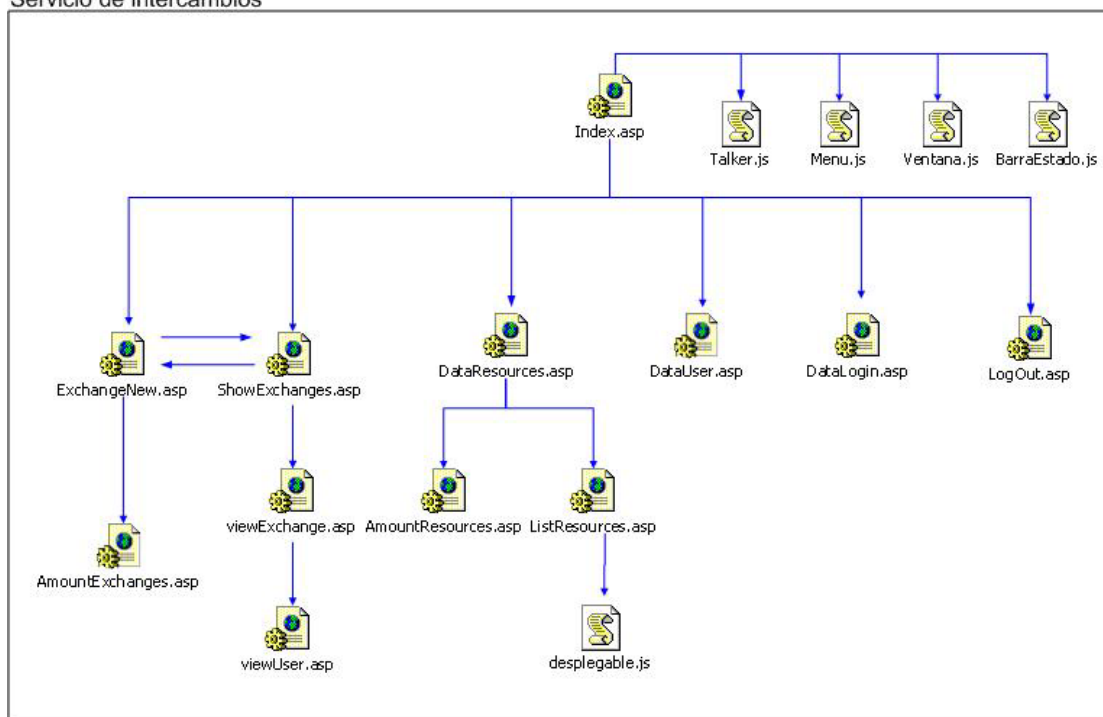
Save Details

4.2 Servicio de gestión de intercambios

En este servicio se muestra todas la posibilidades de un usuario registrado y autenticado en el sistema



Servicio de intercambios





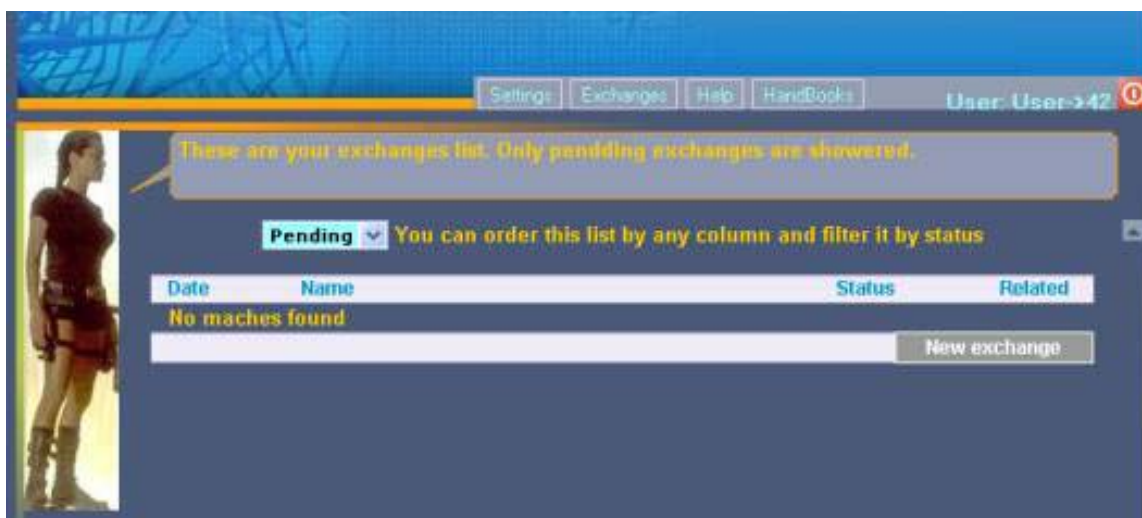
4.2.1 Menú principal

El menú principal de un usuario registrado lo sigue llevando la página *Index.asp*.

Si embargo, la apariencia ha cambiado y las opciones de menú son distintas, disponemos de las herramientas necesarias para que un usuario validado pueda configurar sus preferencias y generar intercambios.

La página principal consta de tres apartados un globo superior en donde se nos mostrara información relevante a las acciones del usuario.

Una imagen de nuestro agente que mostrara la ayuda necesaria a la página en donde estamos simplemente pinchando sobre ella; y por último la pantalla principal de datos.



4.2.2 Modificación de datos

En este apartado se pueden modificar tanto los datos personales como los datos de acceso a la aplicación. Para la modificación de los datos personales se utiliza la página *DataUser.asp*





Para la parte de los datos de acceso se usa *DataLogin.asp*.

Ambas páginas solicitan los datos al usuario y tras comprobar su validez los envían a la BBDD.

4.2.3 Alta de Recursos

En esta sección un usuario podrá introducir en el sistema todos los recursos de que dispone y quiera intercambiar, para ello utilizará la página *DataResources.asp*.

Los recursos que un usuario puede elegir los define el administrador del sistema y están almacenados en la BBDD. (Un usuario puede solicitar el alta de un nuevo recurso a través de un mail al Web master)

La página *ListResources.asp* se los muestra al usuario para que los escoja agrupados por las categorías a las que pertenecen.

En cualquier momento un usuario puede modificar los recursos de que dispone.

Al añadir recursos el usuario deberá de elegir que cantidad quiere. Esta parte la lleva a cabo la página *AmountResources.asp*



Amount: 1
Confirm

4.2.4 Generar intercambios

En este apartado un usuario deberá de decidir que recursos de los que tiene registrados en el sistema quiere cambiar y qué recursos quiere a cambio.

Esta gestión la lleva a cabo la página *ExchangeNew.asp* que solicita al usuario sus preferencias.

Cuando un usuario ya ha hecho su elección y quiere grabar puede optar por crear otro intercambio relacionado con el actual de modo que si uno de ellos se lleva a cabo los otros que se están relacionados se desestimen.

Name* Save Save & Or

Available resources

- FOOD
 - DRINK
 - SOFT DRINKS
 - COKE
 - FANTA
 - ALCOHOL DRINKS
 - FRUIT
 - FISH
 - MEAT
- TOYS
- BOOKS
- GENERAL OBJECT

Resources to Give*

MELONS(1)

Resources to Obtain*

Cuando un usuario quiere incluir un producto al carro deberá de elegir que cantidad del recursos desea. Esto se hace con la página *AmountExchanges.asp*

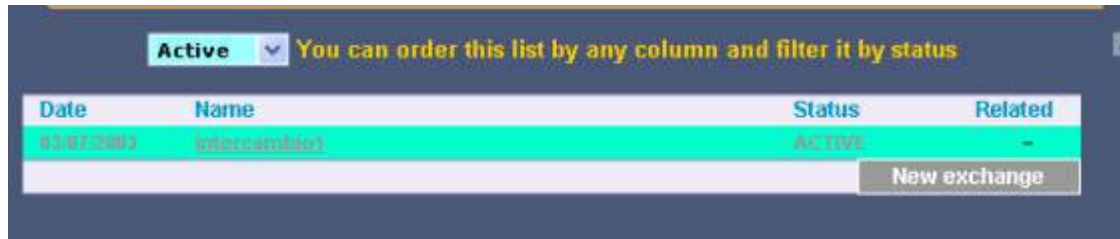
Amount: 1
Confirm



4.2.5 Listado de recursos

En esta sección se dará un listado de todos los recursos que tiene el usuario. Este listado se podrá filtrar para sólo listar las activas, las pendientes de confirmación, etc. Además también se puede ordenar por fecha, estado actual, nombre del intercambio o por relaciones. En este último caso todas las relacionadas aparecen agrupadas con el mismo color.

Toda esta gestión la realiza la página *ShowExchanges.asp*.

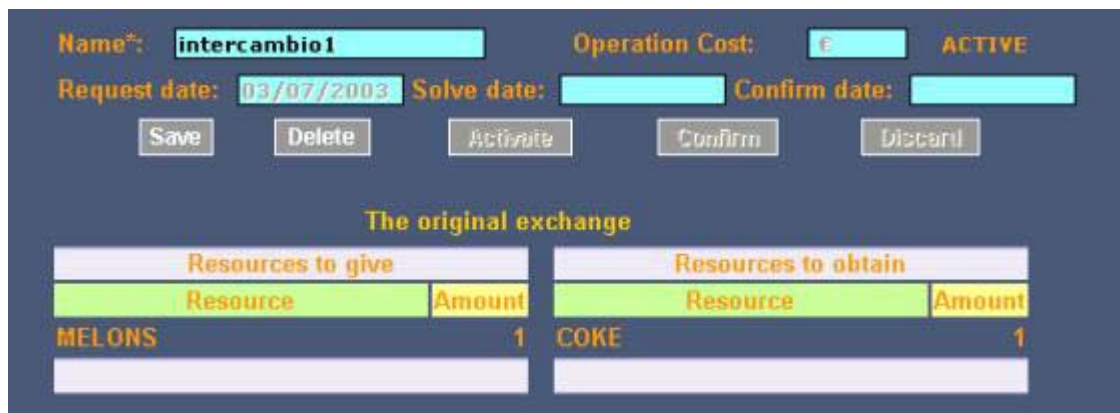


Además tenemos un botón al final del listado para crear nuevos intercambios.

Los nombres de los intercambios son hiperenlaces que nos llevarán a una página para ver el intercambio de forma detallada.

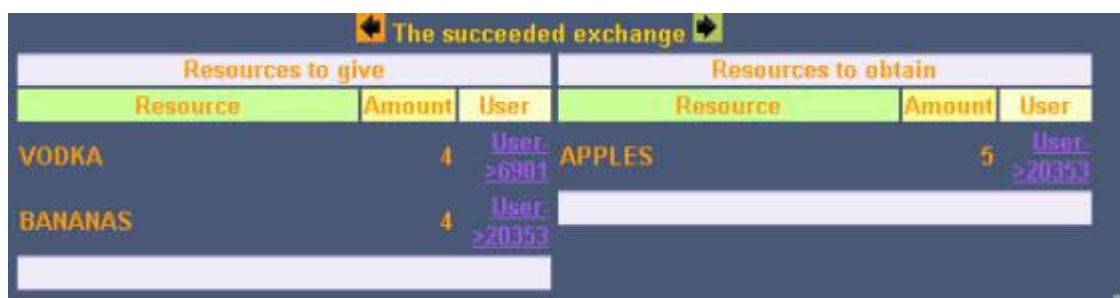
4.2.6 Ver los intercambios

Ahora podemos ver detenidamente un intercambio y el estado en el que se encuentra. La página encargada de ello es *viewExchange.asp* y además de ver el intercambio detenidamente nos permitirá modificar el nombre del intercambio, o borrar el intercambio entero.



Esta imagen corresponde con un intercambio que aún no ha sido realizado.

Si el intercambio ya hubiese sido realizado podríamos ver cual es el intercambio original (una vista como la imagen anterior), y como ha sido finalmente donde se indica que recursos hay que intercambiar y con que usuario.





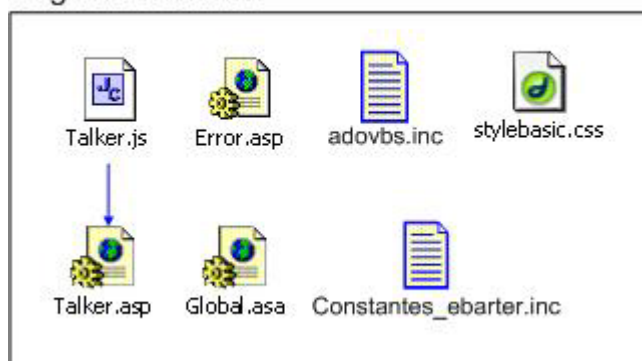
Los nombres de los usuarios son enlaces a la página *viewUser.asp* que nos muestra información del usuario de modo que podamos ponernos en contacto con él.



4.3 Páginas comunes

Por último están las páginas a las que enlazan todas o casi todas las otra páginas.

Páginas comunes



4.3.1 Global.asa

En esta página se definen los procedimientos que se ejecutaran al iniciar la aplicación (entre el primer usuario), iniciar una sesión (cada vez que entra un usuario), finalizar una sesión (cada vez que salga un usuario) y finalizar aplicación (cuando salga el último usuario). Básicamente las utilizamos para definir variables globales como la conexión a la BBDD.

4.3.2 Error.asp

Es una página en la que se muestra que ha habido un error inesperado. Es un metodo de ocultar información al usuario, ejemplo: "error en base de datos SQL Server" que no es relevante para un usuario común o un punto débil ante un usuario peligroso

4.3.3 Talker.asp

Esta página es en la que se muestra la información que nos comunica nuestro agente. Los mensajes que en ella se muestran se los pasa el script *Talker.js*.

Todas las páginas del servicio de gestión de intercambios tienen una referencia a ese script de modo que muestre al usuario la información que se genera en la pantalla (instrucciones de que hacer en cada momento , errores que hayan producido, etc.).



4.3.4 Stylebasic.css

En esta página se definen los estilos de apariencia de la Web. Todos los colores de los textos colores de fondo, tipo de letras ... están definidos en este archivo. De este modo es muy sencillo hacer cambio en la apariencia de la Web.

Este archivo sólo recoge los estilos comunes, i. e si se define un estilo sólo para una página, el estilo irá embebido en la propia página ASP (pej: Index.asp)

4.3.5 Adovbs.inc

Se trata de una página en donde están declaradas todas los constantes definidas por Visual Basic ante una conexión ADO (Active Database Object), constantes para realizar la conexión con la base de datos, con los procedimientos almacenados, etc.

4.3.6 Constantes_ebarter.inc

Hemos reunido ciertas constantes utilizadas durante la ejecución de ebarter, están reunidas todas las frases que la página talker.asp ira mostrándote con nuestra interacción, además de constantes necesarias para el envío de e-mail al Web master, SMPT de salida, etc.

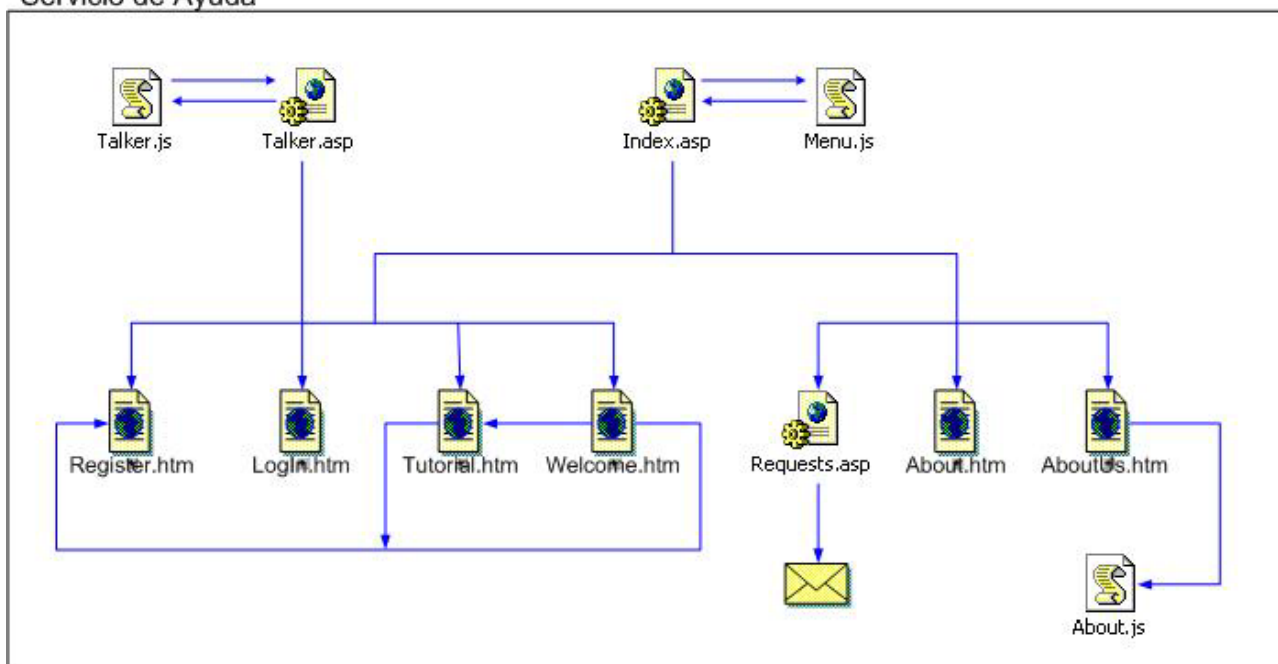


4.4 Servicio de ayuda

El Sitio Web eBarter cuenta con una ayuda online completa y de fácil manejo.

La página Index.asp hace referencia a todas las posibilidades que brinda el sistema eBarter y dentro de cada página contamos con la ayuda inteligente, dependiendo de en que pantalla nos encontremos: creando un intercambio o definiendo nuestros datos personales, pinchando encima de la imagen del agente, nos mostrara una página de ayuda referente al problema que tengamos.

Servicio de Ayuda



También podremos enviar un mail de sugerencias al Web Master desde la página *request.asp*

El formulario de envío de sugerencias se encuentra sobre un fondo azul oscuro. En la parte superior, hay una fila de cuatro botones de radio con las etiquetas: 'New Resource' (seleccionado), 'About Errors', 'Help' y 'Others'. Debajo de estos, hay dos campos de entrada de texto. El primer campo, etiquetado como 'Title:', es un campo de texto simple. El segundo campo, etiquetado como 'Text:', es un área de texto más grande que comienza con el texto 'To WebMaster:'. En la parte inferior del formulario, hay un botón rectangular con el texto 'Send'.



5 Plantillas ASP

5.1 AmountExchanges.asp

Precondición	Se ha seleccionado un recurso en la página de nuevo intercambio,	
Poscondición	Obtenemos un valor para la cantidad del recurso seleccionado en la precondición	
Descripción	Ventana que se abre al añadir un recurso a la mochila en donde defines el numero de recursos que tienes puestos en juego en el intercambio y los que quieres, dicho numero es verificado que este en un intervalo de 0 a 999.9	
Parámetros		
Nombre	Método	Descripción
List	Get	Si se trata de un recurso nuestro o un recurso que queremos
ResourceValue	Get	Identificador del Recurso
ResourceText	Get	Nombre del Recurso
REsourceMax	Get	

* Método = POST/GET



5.2 AmountResource.asp

Precondición	Se ha seleccionado un recurso en la página de añadir recursos a nuestra mochila, es decir lo que queremos,	
Poscondición	Obtenemos un valor para la cantidad del recurso seleccionado en la precondición	
Descripción	Ventana que se abre al añadir un recurso a la mochila en donde defines el numero de recursos puestos en juego en el intercambio, dicho numero es verificado que se trate de un intervalo de 0 a 1000	
Parámetros		
Nombre	Método	Descripción
List	Get	Si se trata de un recurso nuestro o un recurso que queremos
ResourceValue	Get	Identificador del Recurso
ResourceText	Get	Nombre del Recurso
REsourceMax	Get	

* Método = POST/GET



5.3 DataLogin.asp

Precondición	Hemos validado correctamente un usuario, y se ha hecho clic en el menú en login	
Poscondición	Se muestra la información de login: Username, Password, Question, Response	
Descripción	Se trata de acceder a la información de registro de login, que inicialmente se definen cuando te registras por primera vez, pudiendo además modificar los datos.	
Parámetros		
Nombre	Método	Descripción
Login	Post	El valor ya existe y es el que se envía si no es modificado
Oldpwd	Post	El editor no tiene datos, necesita que se recuerde la contraseña inicial para poder modificarla
Pwd	Post	Nueva contraseña en caso de querer cambiarla
Respon	Post	Respuesta a una pregunta generada por el cliente para recordar la contraseña en caso de olvido
Question	Post	Pregunta que al ser respondida se envía un e-mail con el objetivo de recordar la contraseña

* Método = POST/GET



5.4 DataNew.asp

Precondición	Acabamos de descubrir el proyecto ebarter y nos queremos dar de alta como usuario del sistema
Poscondición	Se guarda toda la información solicitada y facilitada por el usuario en una base de datos
Descripción	Se trata de una página de registro, donde el cliente se puede dar de alta para disfrutar de las ventajas del proyecto ebarter.

Parámetros

Nombre	Método	Descripción
NameUser	Post	Nombre del usuario
SurName	Post	Apellidos del usuario
Login	Post	Nombre de usuario
Pwd	Post	Contraseña
Repwd	Post	Repetimos la contraseña por si hay error
Phone	Post	Teléfono
Phone2	Post	Otro posible teléfono
Country	Post	Ciudad
Neighbourhood	Post	Barrio
Address	Post	Dirección
Mail	Post	E-Mail
Respon	Post	Respuesta a una pregunta generada por el cliente para recordar la contraseña en caso de olvido
Question	Post	Pregunta que al ser respondida se envía un e-mail con el objetivo de recordar la contraseña
H1	Post	Elemento oculto que se envía para saber si tiene que guardar la información o solo actualizar el barrio si se ha cambiado de país

* Método = POST/GET



5.5 DataResource.asp

Precondición	Hemos validado correctamente un usuario, y se ha hecho clic en el menú en Resources	
Poscondición	Modificamos los elementos que pertenecen a nuestra mochila, es decir, aquellos elementos que queremos intercambiar o en todo caso se quedara la mochila como estaba antes de acceder a la página	
Descripción	Se trata de una página en donde podemos elegir un determinado recurso del árbol de elementos, una cantidad y se añadiría a nuestra mochila particular de posibles elementos que en un futuro nos gustaría intercambiar	
Parámetros		
Nombre	Método	Descripción
Cart	Post	Se trata del Id que representa al recurso seleccionado
CartAmount	Post	Cantidad del recurso.

* Método = POST/GET



5.6 DataUser.asp

Precondición	Hemos validado correctamente un usuario, y se ha hecho clic en el menú en "Personal Detail"
Poscondición	Se ha modificado la información que se tenía acerca del usuario o en su defecto se mantiene la información inicial
Descripción	Es una página que muestra toda la información personal a excepción de los datos de Login cedidos por el usuario al registrarse para poder acceder al sistema.

Parámetros

Nombre	Método	Descripción
NameUser	Post	Nombre del usuario
SurName	Post	Apellidos del usuario
Phone	Post	Teléfono
Phone2	Post	Segundo teléfono
Country	Post	País
Neighbourhood	Post	Barrio
Mail	Post	e-mail
Address	Post	Dirección
H1	Post	Parámetro oculto que determina si debe guardar la información o recargar un Nuevo barrio si se ha actualizado el país

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 105/133

5.7 Error.asp

Precondición	Cualquier página que haya generado un posible error	
Poscondición	Página en donde se comenta que página ha sufrido un error	
Descripción	Página de información	
Parámetros		
Nombre	Método	Descripción
Error	Get	Se transmite el error
Page	Get	La página que lo ha cometido.

* Método = POST/GET



5.8 ExchangeNew.asp

Precondición	El usuario existe y se ha validado correctamente en el Log in, además ha actualizado su mochila y ha añadido recursos para intercambiar	
Poscondición	Se ha creado un intercambio o grupo de intercambios relacionados, el sistema ha cogido los recursos de la mochila y los ha puesto en juego en eBarter	
Descripción	Esta página permite crear un nuevo intercambio o grupo de intercambios relacionados.	
Parámetros		
Nombre	Método	Descripción
cartToGive	Post	El id que hace referencia al recurso seleccionado
cartAmountToGive	Post	Cantidad del recurso seleccionado
Name	Post	Nombre que el usuario ha puesto al intercambio
OR	Get	Si forma parte de un intercambio condicional

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 107/133

5.9 ListResource.asp

Precondición	Haber accedido a la página de Resources o NewExchange.	
Poscondición	Hemos elegido un identificador de recurso.	
Descripción	Página que nos muestra un menú desplegable con los recursos del sistema ordenados por categorías.	
Parámetros		
Nombre	Método	Descripción

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 108/133

5.10 LogOut.asp

Precondición	Estar validado correctamente en el sistema	
Poscondición	Estamos fuera, offline del sistema en la página de Log In	
Descripción	Es la página que llamamos desde el botón de desconexión con el fin de finalizar la sesión abierta por el usuario.	
Parámetros		
Nombre	Método	Descripción

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 109/133

5.11 LostPasww.asp

Precondición	Hemos fallado el intento de conexión, no nos acordamos de la contraseña	
Poscondición	Accedemos a la página de “LostPasww2.asp”	
Descripción	En esta página pedimos que se introduzca el login del usuario, llamamos a “LostPassw2.asp” pasándole el login	
Parámetros		
Nombre	Método	Descripción

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 110/133

5.12 LostPasww2.asp

Precondición	Ha sido llamada desde “LostPassw.asp”	
Poscondición	Se envía un mensaje con la contraseña del usuario si ha respondido correctamente a su pregunta, en caso contrario, mensaje de error	
Descripción	Recibimos el login de la página anterior y buscamos la información proporcionada por el usuario al registrarse, “pregunta y respuesta” se le hace su pregunta a la espera de que responda correctamente, si es así, se le envía un mail a la dirección que haya especificado con anterioridad, Si el usuario ha introducido un login erróneo, no es advertido como método de seguridad, ante el prueba y error si se intenta hackear la página de este modo, el hacker no sabrá que esta introduciendo mal si Login o respuesta	
Parámetros		
Nombre	Método	Descripción
Login	Get	El login del usuario.
Respon	Post	Respuesta del usuario
question	post	Pregunta del usuario hecha por el sistema, no modificable

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 111/133

5.13 Passw.asp

Precondición	Hemos saltado la presentación o pulsado el botón de Log in en el menú	
Poscondición	Se ha validado correctamente la información y estamos online, hemos equivocado la información y se nos informa de cómo recordad el password	
Descripción	Se trata de una página de control de acceso al sistema, si eres usuario, entras una vez demostrado que lo eres: usuario y contraseña.	
Parámetros		
Nombre	Método	Descripción
Login	Get	Login que escribe el usuario
Pwd	Get	Contraseña del usuario

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 112/133

5.14 Request.asp

Precondición	Usuario bloqueado o quererse poner en contacto con el WebMaster	
Poscondición	El usuario envía un mail al WebMaster	
Descripción	Si un usuario ha sido bloqueado por el administrador del sistema eBarter, se le avisa de ello y si quiere ponerse en contacto con el WebMaster que le explique porque del bloqueo. Se puede acceder en general para enviar cartas a WebMaster, nuevo recurso, otros, etc.	
Parámetros		
Nombre	Método	Descripción
Block	Get	Si es 0 el usuario ha sido bloqueado y preparamos la página para ello
Login	Get	Login del usuario en caso de haber sido bloqueado.

* Método = POST/GET



5.15 ShowExchange.asp

Precondición	Acabamos de validar como usuario del sistema eBarter	
Poscondición	Muestra todos los intercambios creados por el usuario,	
Descripción	Es la página principal de un usuario online, muestra toda la información de los intercambios, pudiendo filtrarlos de diversas formas	
Parámetros		
Nombre	Método	Descripción
Filtro	Get	Que tipo de filtro se ha elegido
Order	Get	Por que campo estamos ordenando, nombre, status, relación, etc.

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 114/133

5.16 Talker.asp

Precondición	Estando online, hemos realizado alguna acción	
Poscondición	Muestra información relevante sobra la acción realizada.	
Descripción	Se trata de una página ideada con el fin de informar al usuario de todas las acciones realizadas por el usuario	
Parámetros		
Nombre	Método	Descripción
Title	Get	Titulo del mensaje a destacar
MSG	Get	Cuerpo del mensaje

* Método = POST/GET



5.17 Viewexchange.asp

Precondición	Hemos seleccionado un intercambio en la página se Showexchange	
Poscondición	Mostramos toda la información referente a el intercambio	
Descripción	Tenemos toda la información que se genera al generar un intercambio y ponerlo en juego en eBarter, Fechas importantes, Coste, estado del intercambio, usuario relacionado si existe, etc.	
Parámetros		
Nombre	Método	Descripción
Exchange	Get	identificador del intercambio
accion	Post	Se refiere al estado del intercambio, Active, Suggestion, discard, etc.
Name	Post	Nombre del intercambio.

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.

Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Diseño de un portal web para la interacción los los usuarios a través de Internet.

Fecha. Creación : 01 Julio 2002

Versión 1.0.0

Sección 5: CLIENTE WEB

Página : 116/133

5.18 Viewuser.asp

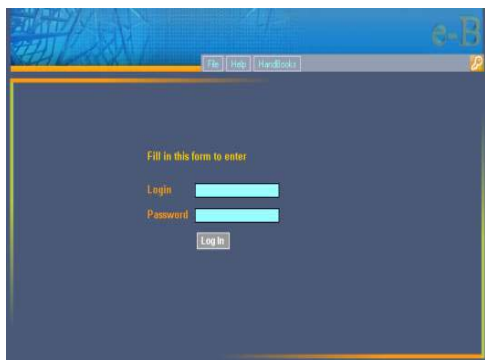
Precondición	Hemos elegido un intercambio solucionado o en espera de confirmación	
Poscondición	Se muestra la información del usuario dispuesto a intercambiar	
Descripción	Nos muestra la información de un usuario que tiene lo que queremos y quiere lo que damos.	
Parámetros		
Nombre	Método	Descripción
Login	Get	Login del usuario dispuesto a intercambiar

* Método = POST/GET



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.
Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter
Descripción : Manual de usuario del portal web.
Fecha. Creación : 01 Julio 2002 Versión 1.0.0
Sección 6: **MANUAL DE USUARIO WEB** Pagina : 117/133



SECCION 6. : MANUAL DE USUARIO WEB



1 Introducción

La Web eBarter no contiene ninguna complejidad que vaya mas allá de la navegación de la Web, es cierto que hay que estar familiarizado con el funcionamiento para agilizar la formalización de intercambios.

Intentaremos que el manual siga los posibles pasos que haría un nuevo usuario al entrar en eBarter.

En todo momento de nuestra navegación por eBarter dispondremos de un menú con todas las herramientas posibles de trabajo en eBarter.

Menú para usuario Offline:



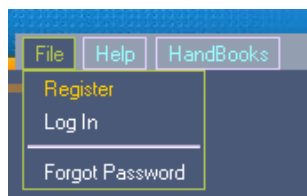
Menú usuario Online:



2 Registro:

Para ser usuario de e-Barter tendrás que registrarte rellenando un formulario con tus datos de acceso y con algunos datos personales. El registro no te llevará mas que unos pocos segundos y una vez registrado podrás empezar, de forma inmediata, a utilizar el servicio.

En el menú inicial en la parte superior de la pantalla seleccione la pestaña File y posteriormente Register.



Le aparecerá un formulario como el siguiente

Fill in this form with your details

Login*	<input type="text"/>	Question	<input type="text"/>
Password*	<input type="text"/>	Response*	<input type="text"/>
re-password*	<input type="text"/>		
<hr/>			
Name*	<input type="text"/>	Phone*	<input type="text"/>
Surname*	<input type="text"/>	Phone2	<input type="text"/>
e-mail*	<input type="text"/>	Country	<input type="text" value="BELGIUM"/>
Address	<input type="text"/>	Neighborhood*	<input type="text"/>
<input type="button" value="Save Details"/>			



En él deberá de rellenar los datos de acceso Login y password además de una pregunta y una respuesta por si se olvidase del password. También deberá de rellenar un apartado correspondiente a los datos personales, donde deberá de indicar entre otros su nombre completo, e-mail (muy importante que sea válido), teléfono y dirección.

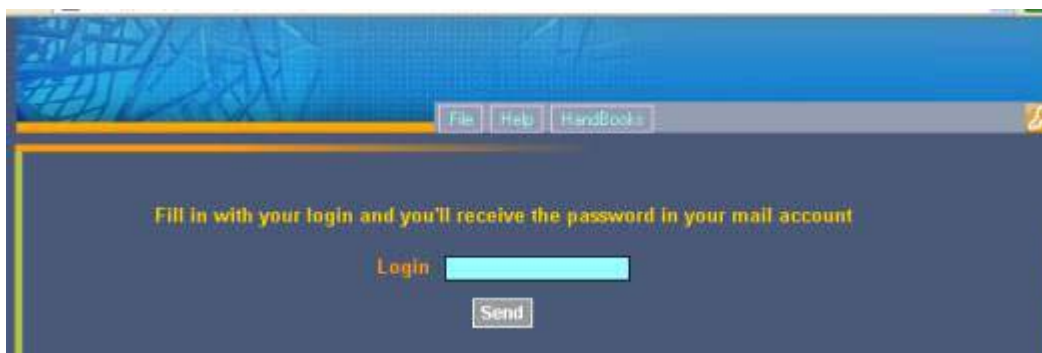
Una vez completados los datos pulse el botón "Save". Si ha completado todo correctamente tendrá acceso inmediato a su página inicial.

3 Login

Esta es la pagina de validación de usuario y contraseña para acceder al sistema e-Barter y a tu Información personal, para ello previamente debes estar registrado.

Si no es correcta la validación, el sistema mostrara un error dándote a acceso a recordar tu Contraseña.

Una vez hemos pinchado en "here" se nos pedirá el login para enviarnos la contraseña a la dirección de mail que hayamos facilitado en el registro.



Si por el contrario, hemos validado correctamente la información, entonces accederemos a la pagina principal del sistema con nuestra información.

En este apartado vamos a intentar familiarizarle con el aspecto de la pantalla.

Cuando entra en el sistema como usuario registrado le aparecerá la siguiente pantalla



En ella se pueden distinguir varias partes.

- ✎ [Menú principal de la aplicación.](#)
- ✎ [Agente del sistema](#)
- ✎ [Pantalla de datos.](#)

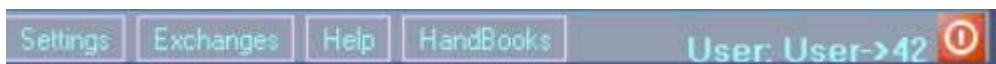
Veamos cada una de ellas:



4 Menú principal

En la parte derecha del menú tenemos el nombre del usuario que está registrado y un botón para salir del sistema.

En la parte izquierda podemos ver las opciones del menú.



4.1 Menú Settings

En la opción "[Personal Details](#)" podrás modificar nuestros datos personales.

En la opción "[Access Details](#)" podrás modificar tus datos de acceso.

En la opción "[Resources](#)" podrás definir los recursos de que dispones para intercambiar.

4.2 Menú Exchanges

En la opción "[New Exchange](#)" podrá crear un nuevo intercambio.

En la opción "[View Exchange](#)" verás un listado de todos los intercambios que tienes en el sistema

En la opción "[View User](#)" veras la información de un usuario que quiera realizar la transacción propuesta



5 Agente del sistema

El objetivo del agente es ayudarte en caso de tener algún problema con la aplicación, darte sugerencias de que debes hacer en cada momento y alertarte los errores que hayas cometido.

El agente consta de una imagen con tu agente personal (a la izquierda de la pantalla) un cuadro bocadillo donde podrás leer lo que vaya diciendo en agente.

En cada pantalla que visites el agente te dirá que debes de hacer y con que objetivo y en caso de que haya algún error también te avisará de este hecho.

Además puedes pedir al agente mayor detalle en lo que debes de hacer en la pantalla pulsado sobre él.

En este ejemplo podemos ver como al darle a "Update" sin introducir el campo password nos dice que es obligatorio rellenar todas las casillas señaladas.

Además podemos ver como al poner el cursor sobre el agente cambia al cursor de ayuda.

En caso de hacer click sobre el agente nos aparecerá la pantalla de ayuda correspondiente a la modificación de los datos de acceso.



6 Pantalla de datos

- [Datos Personales](#)
- [Datos de Acceso](#)
- [Recursos](#)
- [Nuevo Intercambio](#)
- [Ver Intercambio](#)

6.1 Datos Personales

Name*	<input type="text" value="Nombre"/>	Phone*	<input type="text" value="55555555"/>
Surname*	<input type="text" value="Apellidos"/>	Phone 2	<input type="text"/>
e-mail*	<input type="text" value="email@correcto.com"/>	Country*	<input type="text" value="SPAIN"/>
Address	<input type="text" value="Direccion"/>	Neighborhood*	<input type="text" value="MADRID (CEN)"/>
<input type="button" value="Update"/>			

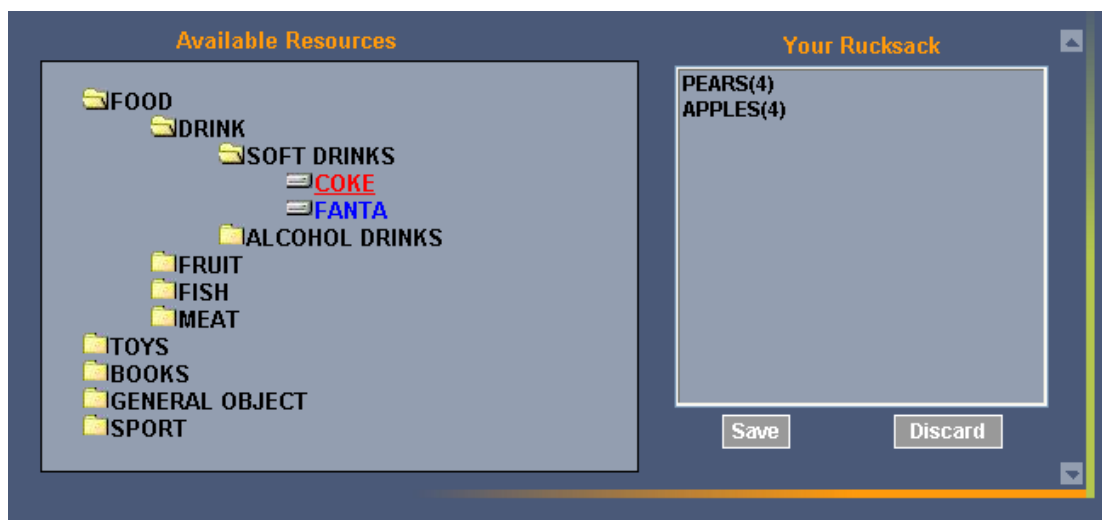
6.2 Datos de Acceso

Login*	<input type="text" value="Login"/>	Old password*	<input type="password" value="••"/>
New password*	<input type="password"/>	Question	<input type="text" value="Pregunta relevante"/>
re-password*	<input type="password"/>	Response*	<input type="text" value="Respuesta secreta"/>
<input type="button" value="Update"/>			

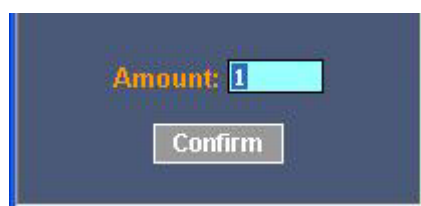


6.3 Recursos

La selección de recursos nuestros es esencial en el funcionamiento de eBarter, pues debemos comunicar al sistema que recursos disponemos antes de poder empezar a generar intercambios, no puedo cambiar peras por algo si el sistema no sabe que tenemos peras.



Para elegir recursos simplemente desplegamos el árbol, elegimos producto, entonces nos pedirá la cantidad de elementos de ese recurso



la definimos y se añadirá sola a nuestra mochila, solo queda pulsar el botón de "Save" para aceptar.

Si acabamos de registrarnos veremos vacía la pagina en donde se mostraran los intercambios que hayamos creado.

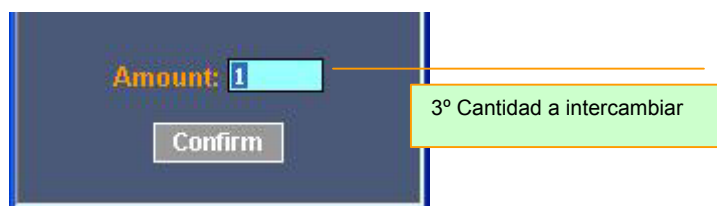
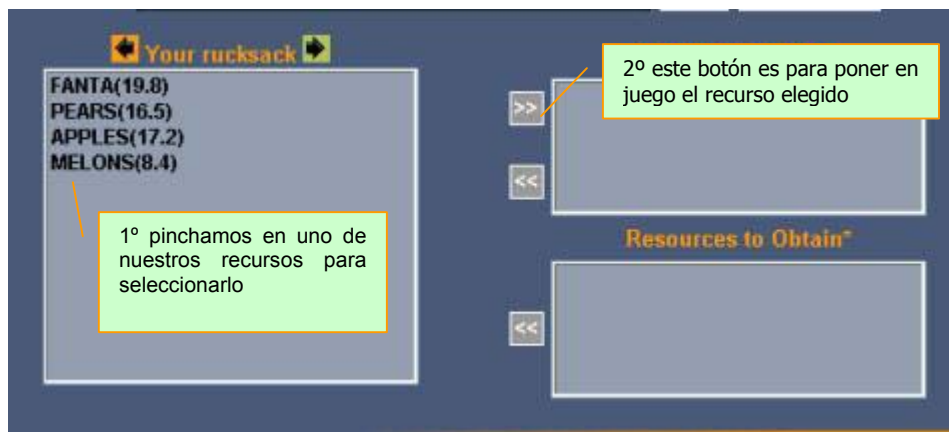
Para crear un nuevo intercambio previamente debemos asegurarnos haber configurado la mochila recursos, es decir, aquellos elementos que disponemos y queremos intercambiar



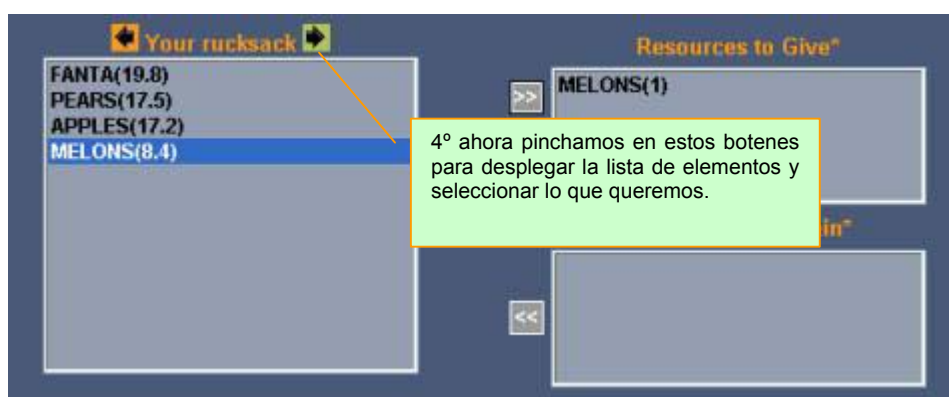
7 Intercambios (Exchanges)

Una vez tenemos completa la mochila con nuestros recursos, podemos empezar a realizar la gestión de los intercambios que queremos realizar. Lo primero que debemos hacer es crear nuevos intercambios que el sistema pueda evaluar, para ello pinchar en el menú, "New Exchange"

7.1 New Exchange



Ya tenemos elegido nuestro recurso o recursos a intercambiar





Ya tenemos la lista productos elegimos lo que queremos igual que añadimos los recursos a la mochila

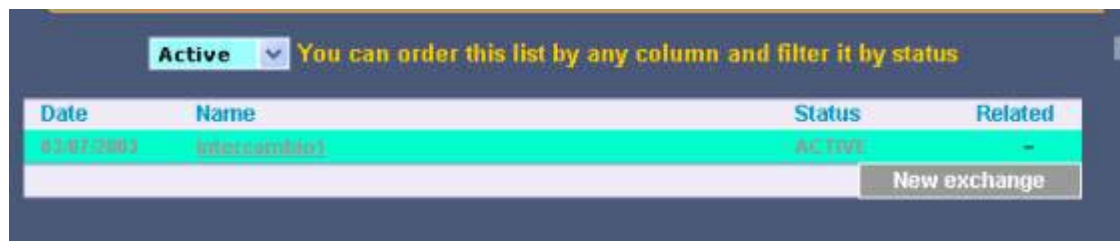


Por ultimo le damos un nombre al intercambio y salvamos. "Save"

También podemos generar intercambios relacionados, es decir, tu puedes querer intercambiar una coke por una fanta, pero te puede dar igual pedir un vodka, en lugar de crear dos intercambios como solo tenemos una coke crearíamos un intercambio relacionado.

7.2 View Exchanges

Ya disponemos de información en la pagina principal. Para ver la evolución del intercambio pinchamos en el intercambio que queramos ver.



La siguiente pantalla muestra toda la información con respecto al intercambio Nombre, coste del intercambio, fechas importantes y recursos en juego.





En el estado del intercambio veremos en que situación se encuentra ahora mismo esta activo esperando encontrar otro usuario con el que realizar la transacción.

En el caso de haber encontrado un usuario con el que negociar cambiaria el estado deberíamos confirmar o descartar el intercambio según nos interese.

Name: intercambio1 **Operation Cost:** € **EBARTER SUGGESTION**

Request date: 03/07/2003 **Solve date:** **Confirm date:**

Botones de información del posible intercambio

The original exchange

Resources to give		Resources to obtain	
Resource	Amount	Resource	Amount
MELONS	1	COKE	1

Nuevo Estado

Podemos ver lo que realmente hemos dado y obtenido

The succeeded exchange

Resources to give			Resources to obtain		
Resource	Amount	User	Resource	Amount	User
VODKA	4	User: >6901	APPLES	5	User: >20353

Accedemos a la ventana de información del otro usuario

7.3 View User

Bien en esta ultima ventana, podemos ver los datos del usuario con el que realizar la transacción.

User User->6901

Country: BELGIUM **Neighborhood:** MADRID (CENTER)

e-Mail: **Phone:**



Autores : Javier Niño Fernández
Rubén Torresano Rubio
Rene Alonso Peña Sánchez
Tutor Proyecto : Manuel Nuñez.
Proyecto S.I. / e-Barter. (Año 2002/03)

Título Documento : Memoria del proyecto e-Barter

Descripción : Anexos a la documentación.

Fecha. Creación : 01 Julio 2002

Sección 7: **ANEXOS**

Versión 1.0.0

Página : 128/133



SECCION 7. : ANEXOS.



1 Requisitos del sistema.

El sistema de Intercambios eBarter ha sido generado con las ultimas novedades en implementación cliente servidor, esta completamente orientado a Windows puesto que no se dio ningún tipo de especificación inicial y por supuesto no se pidió multiplataforma.

No se trata de un sistema con fines comerciales por lo menos por ahora, sus fines están orientados a la enseñanza, por tanto, crear un sistema multiplataforma retrasaría el desarrollo real del mismo.

Consta de una Base de datos realizada en SQL Server que da soporte a todo el sistema tanto clientes como administrador

1.1 Servidor:

El servidor o servidores que darán soporte al sistema los cuales deben estar preparados para ello.

El sistema operativo para el cual esta optimizado el funcionamiento por la configuración estándar del mismo es Windows 2000 y 2003, esto no implica que no funcione en NT la única diferencia es que se tendría que actualizar el Service Pack de NT y actualizar el IIS a la ultima versión.

La base de datos se diseño en SQL Server, para una mayor compatibilidad con el desarrollo, puesto que se decidió optimizar para Windows.

Por tanto, es necesario la instalación de dicha base de datos en el Sistema.

El motor del sistema se decidió que lo desarrollaríamos en Java por ser un lenguaje Puro y multihebra, por tanto, si se necesita compilar el programa fuente hace falta instalar la versión de JDK 1.3 como versión mas antigua, pues las librerías utilizadas no son estándar hasta esa versión.

La pagina Web fue desarrollada en ASP utilizando los métodos proporcionados en la versión IIS5, una versión anterior no seria capaz de interpretar correctamente el código.

Por ultimo el programa Administrador fue realizado en Delphi 7, se buscaba un lenguaje estable, fuerte y funcional además de orientado a Windows en diseño, que implica la configuración de el BDE de Delphi para la conexión con Base de Datos.

1.2 Cliente:

El ordenador cliente solo esta pensado para la interacción del usuario con el sistema a través de la Web. Sin embargo, dicha Web, para un diseño dinámico, sencillo y cercano a los programas de Windows se utilizo tecnologías no inferiores a la versión 5.5 de IE.

Tanto la web que proporciona la interfaz con el usuario como el programa de administración han sido diseñados para una resolución mínima de pantalla de 800x600



2 Manual de Instalación del Sistema eBarter.

La instalación del sistema eBarter es un desarrollo de pasos no demasiado complejos pero tal vez esenciales para el correcto funcionamiento del proyecto.

La no configuración o salto de pasos puede llevar al mal funcionamiento o mala ejecución del sistema.

La conexión entre los diversos sistemas se realizan de forma diversa, no hemos querido estandarizarlo debido a la diferencia de eficiencia que puede ocasionar en cada uno de los programas cliente. Por tanto veremos como el programa administración de conecta a la Base de datos por medio de BDE, el motor Java por ODBC y la Web físicamente.

El Servidor será quien sufrirá toda la instalación de herramientas necesarias para la ejecución del sistema como edición del mismo.

2.1 Desarrollo de la instalación

- 1 [Sistema Operativo.](#)
- 2 [Internet Information Server \(IIS5\)](#)
- 3 [ASPMail](#)
- 4 [Modificación de Global.asa](#)
- 5 [SQL Server](#)
- 6 [BDE Borland Delphi7](#)
- 7 [JDK 1.3*](#)
- 8 [Setup eBarter 2000.exe](#)

2.1.1 Sistema Operativo:

Como hemos mencionado en el apartado anterior, es recomendable la instalación de Windows 2000 como sistema operativo base del sistema.

2.1.2 Internet Information Server (IIS5)

Es quien se encarga de interpretar el código ASP de la Web, además de difundir la Web al cliente. No puede ser una versión anterior a la 5, pues no interpretaría correctamente el código ASP. La Web se guarda en la dirección siguiente:

C:\inetpub\wwwroot\eBarter

Esto implica que todo aquel que este conectado con el servidor en red, con la siguiente dirección URL accederá al sistema.

http://nombre_server_o_IP/eBarter



2.1.3 ASPMail

La pagina Web es capaz de enviar e-mail de información de contraseña o al WebMaster si es una consulta, para ello es necesario instalar las DLL contenidas en el programa ASPMail.

La configuración de salida SMTP de la Web esta guardada en un archivo llamado Constantes_eBarter.inc en donde se encuentra la Web.

2.1.4 Configuración de Global.asa

Para la conexión de la Web con la Base de Datos es posible de generarla de dos formas, ODBC o física.

A pesar de que el motor Java utiliza el ODBC y la Web podría utilizarlo también, hemos decidido hacerlo físicamente por seguridad y velocidad.

Para ello se debe editar el fichero Global.asa y modificar el campo de llamada a la base de datos en donde se refleja el nombre del servidor y nombre de usuario, contraseña de acceso a la BD.

2.1.5 SQL Server

Como hemos dicho hace falta la instalación de SQL Server que es el soporte de todo el sistema, por supuesto es necesario la instalación de la propia base de datos, hemos generado un Script el cual te genera la base de datos entera, para ello hay que abrirlo desde el "administrador de Consultas SQL" y ejecutarlo.

2.1.6 BDE Delphi

El BDE (Borland Database Engine) es el software que hace de interfaz entre el programa de administración y la BD, permitiendo la conexión e interpretación de las ordenes dadas al RDBMS. Por esta razón para la utilización del programa de administración se requiere la instalación y configuración de este software

Se requiere generar una conexión llamada "**eBarter_BDE_SQL**" definiendo correctamente los campos necesarios para la conexión con la BD. (ver mas detalles en la sección referente al programa de administración)

2.1.7 JDK 1.3

En el caso de querer editar el código fuente del motor Java.

2.1.8 Setup_eBarter_2000.exe

Es el instalador que hemos generado para guardar toda la información con respecto a eBarter e incluye :

- Programas (Ejecutables del sistema y software adicional necesario)
- Web
- Configuración de ODBC,
- Configuración de eBarter como pagina de inicio.



3 Glosario de términos

Barter: Cuando un intercambio da a otro intercambio una determinada cantidad de un recurso de crea un barter.

Colaboración: Varios intercambios pueden establecer una colaboración para intentar encontrar una solución. Esta colaboración se establece poniendo todos los recursos en común de forma que se quiere conseguir la unión de los recursos que querían conseguir por separado y lo que se puede ofrecer también es la unión de que lo ofrecían por separado.

Exchange: Ver Intercambio.

Intercambio: Un intercambio lo crea un usuario con el fin de lograr algunos recursos. A su vez debe de ofrecer un conjunto de recursos que está dispuesto a dar para conseguir los que quiere.

Transacción: Es cambio de recursos entre dos o más intercambios con el fin de satisfacerse mutuamente. Una transacción está formada por un conjunto de barterers o trueques de forma que se forma un grafo dirigido y conexo de relaciones entre los intercambios.

Transacción aplanada: Una transacción se dice aplanada cuando ninguno de los barterers que la forman implica a una colaboración.

Transacción virtual: Durante el proceso de búsqueda se toman numerosas decisiones y muchas de ellas provocan finalmente un backtraking. Una transacción es virtual hasta el momento que se confirma que la rama elegida es correcta y no se va a hacer backtraking. Una vez que esto ocurre la transacción virtual para a ser una transacción y además se aplanan.

Trueque: Ver Barter.



4 Bibliografía

A continuación se detalla la bibliografía utilizada durante el desarrollo de este proyecto

- The design patterns Java Companion. James W.Cooper. Ed: Addison-Wesley 1998.
- Thinking in Java, 2nd edition Revision 12.
- <http://java.sun.com>
- <http://programación.com>
- MSDN 6.0.
- Active Server Pages 3.0. Mitchell y Atkinson. Ed: Prentice Hall
- A fondo Microsoft SQL-Server version 6.5. Ron Soukup, Ed. McGraw Hill (Microsoft Press)
- Microsoft SQL Server 7.0 System Administration. Ed. McGraw Hill
- Manuales de referencia de SQL Server 2000

Además de la bibliografía especificada se ha obtenido información requerida para el desarrollo del proyecto a través del uso de Internet y la consulta de diferentes páginas web especializadas en los diferentes temas abordados (desde conceptuales a técnicos)